ROBUST UPPER BODY POSE RECOGNITION IN UNCONSTRAINED ENVIRONMENTS USING HAAR-DISPARITY

A thesis submitted in partial fulfilment of the requirements for the Degree of Master of Computer Science and Software Engineering in the University of Canterbury

by Chu, Cheng-Tse

Supervised by Richard Green

University of Canterbury

2008

Abstract

In this research, an approach is proposed for the robust tracking of upper body movement in unconstrained environments by using a Haar-Disparity algorithm together with a novel 2D silhouette projection algorithm. A cascade of boosted Haar classifiers is used to identify human faces in video images, where a disparity map is then used to establish the 3D locations of detected faces. Based on this information, anthropometric constraints are used to define a semi-spherical interaction space for upper body poses. This constrained region serves the purpose of pruning the search space as well as validating user poses. Haar-Disparity improves on the traditional skin manifold tracking by relaxing constraints on clothing, background and illumination. The 2D silhouette projection algorithm provides three orthogonal views of the 3D objects. This allows tracking of upper limbs to be performed in the 2D space as opposed to manipulating 3D noisy data directly. This thesis also proposes a complete optimal set of interactions for very large interactive displays. Experimental evaluation includes the performance of alternative camera positions and orientations, accuracy of pointing, direct manipulative gestures, flag semaphore emulation, and principal axes. As a minor part of this research interest, the usability of interacting using only arm gestures is also evaluated based on ISO 9241-9 standard. The results suggest that the proposed algorithm and optimal set of interactions are useful for interacting with large displays.

Acknowledgements

I would like to acknowledge and extend my heartfelt gratitude to the following persons who have made the completion of this master's thesis possible:

To Jenny, Chris and Nancy, for love and support
To Wan, for understanding and for heart
To Sharon and Philip, for support and inspiration
To Dr. Richard Green, for continuous support and vital encouragement

I would not have reached here without you

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Introduction

In the last decade, there has been a growing interest in the development of systems and techniques for people detection and tracking particularly in public interactive environments (Figure 1). Robustly tracking human pose in such an environment is challenging because constraints on clothing, illumination and background must be relaxed. Pose ambiguities of the complex articulated structure of the body are amplified by the anthropometric variations of somatotype (from ectomorph to endomorph), age, gender and race. Furthermore, a problem arises when an interactive display is so large that the entire display surface cannot be reached, making the use of touch screen technology impossible.

Most of the early approaches to human body tracking relied on magnetic, joint markers [1] or touch screens. However in public spaces displays, a marker-free computer vision approach is also needed.

Figure 1. A public interactive environment

Tracking skin colour regions of face and hands is a recent popular approach to acquiring upper body pose. Unfortunately this approach is vulnerable to illumination variations causing changes to the hue values from digital cameras. Although skin manifold tracking has been proven to be robust in laboratory environments where clothing, lighting, background, occlusion and reflectance [2] could be controlled, it is not suited to public spaces with changing light conditions and where clothing cannot be constrained to reveal only the face and hands.

People tracking based on monocular images is a well explored topic, approached in most of the cases by the integration of multiple visual cues. However, nowadays, the use of stereo vision for these purposes is an active research area. The availability of commercial hardware and software to solve the low-level problems of stereo processing, as well as the lower prices for these devices, makes them an appealing sensor to be used in interactive systems. Stereo vision has several advantages over monocular systems. First, all the algorithms designed for monocular images can be applied, but with additional depth information. Depth information can be employed to achieve a better background segmentation, tracking of people and a better understanding of their gestures. Second, disparity information makes the systems more robust against illumination changes. Therefore, this thesis proposes an algorithm that employs stereo vision and is robust in real scenarios where illumination changes might occur.

1.2 Algorithm overview

Many human-computer interaction techniques exist which use arm gestures, motion or a combination of both[3-11]. Different systems have also been proposed to recognize these gestures and motion [12-19]. However these systems rely on training or a 3D visual-hull reconstruction

which are complex and impractical for an interactive environment. In this research, a new method is proposed which does not rely on training or reconstruction of the 3D object. It consists of stereopsis, a cascade of classifiers for face recognition, anthropometric constraints, orthogonal projections, closest component matching algorithm and closed world tracking to robustly track upper limb pose and movements in an unconstrained environment.

Figure 2 shows the overview of the proposed system. Starting from the top left corner, a stereo camera is used to acquire image pairs of the scene. Then it is passed into a cascade of boosted Haar classifiers to locate any frontal human face on the left image. Stereopsis is used to calculate stereo depth and 3D point cloud of the whole scene. These 3D points are transformed according to the camera orientation. 3D location of the head and anthropometric constraints are used to define an interaction volume for this user. Interest points are searched within this volume to establish the user's pose. The user's pointing direction is derived by finding the hand eye vector, which is a line from the eye to the furthest point from the face (the tip of the finger). In order to identify arm and hand poses and tracking arm positions, the 3D depth data are projected into three orthogonal planes: namely XY, XZ and YZ planes. Morphological filters and median filters are applied to these three images to improve tracking. Arm orientations are located by identifying and calculating the orientations of these blobs on the projection images. Finally this information is tested in the proposed applications using the emulation of the existing Iphone/Ipod touch direct manipulation and flag semaphore recognition.

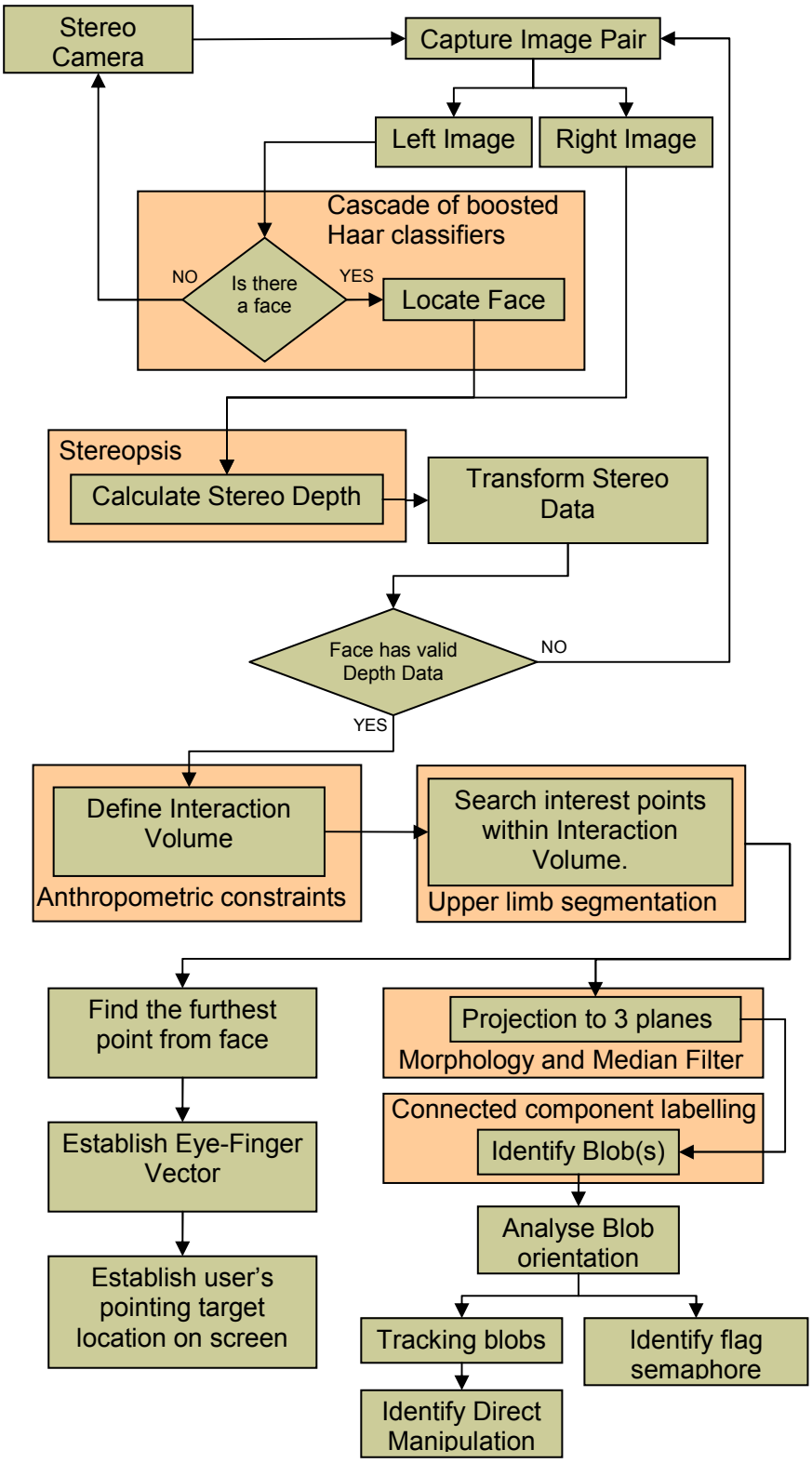


Figure 2. System overview.

Apart from presenting a novel algorithm for upper limb pose recognition, this thesis also presents interaction studies on modern screen interactions and proposes a complete optimal set of screen interactions for public interactive environments.

1.3 Chapter layout

The layout of this thesis is intended to first introduce broad topics and prior research and technology to the reader, providing material to underpin the main focus of the work. This is split into two major focus areas: the novel algorithm presented in this thesis, which allows robust upper limb tracking in an unconstrained environment; and a proposed complete optimal set of modern screen interaction studies for a public interactive environment.

In chapter 2, different large-scale interactive screen technology will be first presented, followed by details of the markerless computer vision solutions employed for limb tracking in unconstrained environments including colour cue based detection. Stereo vision based limb tracking is then presented as it is employed in this research to overcome the limitation of using colour cue detections. Gesture based interactions will also be presented at the end of chapter 2, providing background understanding to modern screen interactions.

In chapter 3, a detailed description of the proposed algorithm for upper limb tracking in an unconstrained environment is presented.

In chapter 4, the focus will be on the proposed complete set of direct manipulative gestures for large screen interactions which have been

implemented in the current limb tracking system and are based on the algorithm presented in chapter 3.

Chapter 5 presents the quantative results and subsequent analysis of the proposed upper limb tracking system. They suggest that the proposed algorithm and optimal set of interactions are useful for interacting with large displays.

While the main focus of this research is on robust upper limb tracking in computer vision, chapter 6 presents a usability study – an evaluation of arm pointing using the ISO 9241-9 standard.

Chapter 7 provides a conclusion detailing the goals achieved and knowledge gained from this research, and suggests possible further directions which may be explored.

This thesis concludes with a bibliography, and survey forms used in usability studies and colour cue detection formulas are included as appendices.

2. Background

2.1 Introduction

Tracking people and their body parts in sequences of images present challenging and difficult problems in computer vision and have been well studied in the last two decades. Recently, methods based on stereo vision have attracted increasing attention since 3D information can be exploited. In this research, we focus on face and upper limb tracking in a public interactive display environment using a single stereo camera pair, where a markerless and non-colour cue dependent approach is necessary.

The following sections outline the taxonomy of public display environments, followed by literature reviews of markerless tracking. The colour based detection technique is discussed in section 2.4 and an overview of model and feature based tracking is given in section 2.6. The taxonomy of gesture interactions is provided in section 2.7.

2.2 Projection Methods for Interactive Displays

As the vision of ubiquitous computing edges towards reality, large displays are being increasingly used in public, semi-public, and private spaces such as airports, meeting rooms, design studios, research labs and homes. Beyond simply broadcasting information to individuals, this also raises an opportunity for the development of interactive displays to serve human to human, and human to computer interactions. Examples would be collaborative work, exchanging data, publishing information and advertisements. With the advent of novel sensing and display technologies, interactive systems are able to move the input and display capabilities of computing systems on to everyday surfaces such as walls

and tables. These efforts are often conducted in the spirit of ubiquitous computing research, where the goal is to make computing resources accessible, seamless, distributed and immediate. Such systems pose interesting challenges for interaction design, signal processing and engineering.

In the following, the six most common projection methods employed in interactive systems today are described (Figure 3). The simplest is front projection (FP). The projector is mounted along the normal axis in front of the screen, but anyone who stands between the projector and the screen will occlude the projector outputs and produce shadows on the screen. A warped front projection (WFP), as used by the 3M IdeaBoard [20], and the Everywhere Display Projector [21], is introduced to minimize occlusion by having a projector mounted off the normal axis. But projection materials need to be processed so that the output is warped to provide a corrected display on the screen. In a passive VRP (PVRP), two front projectors are mounted on opposite sides relatively to the normal axis to redundantly illuminate the screen. This reduces the number, size and frequency of occlusions. However, users standing very close to the screen may still completely occlude portions of the output but usually only occlude the output of one of the projectors, resulting in "half-shadows" where the output is still visible at a lower level of contrast. Similarly to PVRP, active VRP (AVRP) adds a camera or other sensor which determines when one of the projectors is occluded. The system then attempts to compensate for this occlusion by boosting output power from the other projector(s) to increase contrast in the "half-shadow" area(s) [22, 23]. In addition to AVRP, AVRP-BLS adds the ability to detect and turn off projector output that is shining on an object other than the screen, such as an intervening user. This blinding light suppression allows users to comfortably face the projectors without

bright light being projected into their eyes or onto their bodies [24]. Rear projection (RP) systems use a single projector mounted behind the screen, a rear projection solution prevents occlusions and shadows completely, but requires extra, dedicated space for the beam path. Even in a new construction, rear projection is an expensive option.

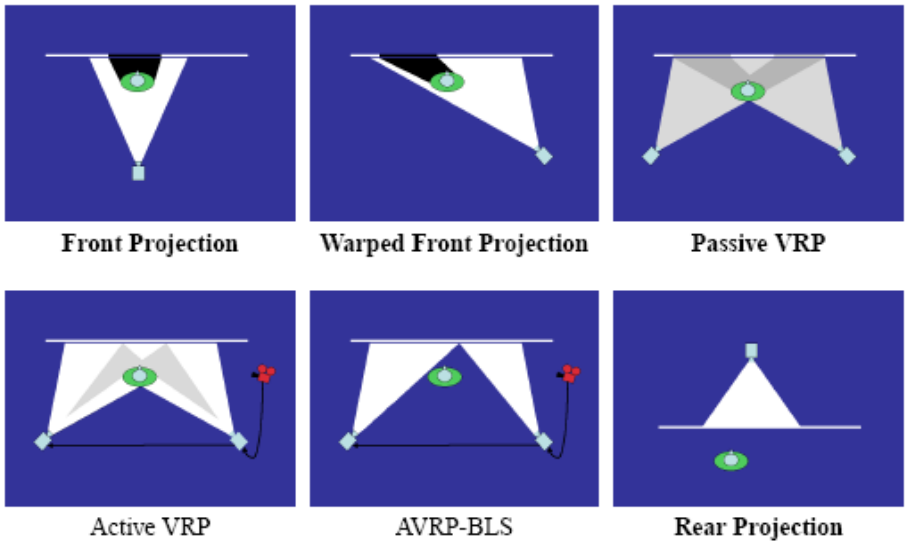


Figure 3. projection technologies

Owing to the simplicity of the front projection method as described earlier, and the fact that it is the most commonly used projection method in public spaces, we consider in this research, the case of interactive environments with large displays on the wall using this method. The interactions in this environment are usually frontal interactions; in another words, a user normally would interact with the display only when facing it. Then one of the main problem domains of this research arises, which is how to robustly track frontal upper body movement. This thesis proposes a novel algorithm to overcome this problem domain. The detail is explained in chapter 4.

2.3 Tracking Systems Survey

2.3.1 Markerless Tracking

The problem of markerless tracking of the human pose has attracted the attention of many researchers over the past few years. Many different approaches exist as can be seen in the following literature, but can be classified by the number of cameras they use: monocular or multi-view.

Monocular approaches have the advantage that they work at low cost and have a very simple hardware setup. This makes them particularly preferable for surveillance applications where a multi-camera calibration may not be applicable in many cases [25-28].

There are many multi-view approaches for markerless body tracking. Gavrila and Davis [29] use 4 calibrated cameras. The cost function is calculated from extracted edges in the neighbourhood of the target contour predictions. Then a robust variant of the Chamfer distance is computed. The formula for computing Chamfer distance is shown in equation 1, where the Chamfer distance $d_{a,b}(p,q)$ between two points, p and q can be calculated. k_a and k_b represent the number of a- and b-moves on the shortest path from p to q.

$$d_{a,b}(p,q) = k_a a + k_b b \qquad (1)$$

Gavrila and Davis's model has 22 DOFs; the estimation is processed by recursive search space decomposition: using a best-first search, the best torso/head configuration is found, then the arm. The main limitation of this approach is that the joints and sizes constraints of the human body are not incorporated. Hence fitting body parts fails when the body parts are too close together which causes problems in proximity segmentations.

Furthermore, the use of tight-fitting clothes, with sleeves of contrasting colours, makes the segmentation easier. In our proposed tracking system, the user can wear normal loose fitting clothes.

Kakadiaris and Metaxas [30, 31] have developed an algorithm to acquire the shape of a human body surrounded by three orthogonal cameras. The body was modelled by a set of deformable shapes which are used to perform 3D body tracking. An extended Kalman filtering is employed for the model prediction. Equation 2 shows the formula of this extended Kalman filter. Where $q(t_k)$ is the vector of state variable, $h_k(q(tk))$ is a nonlinear function which relates the input data to the model's state and $g = D^{-1}(f_q + f_c)$. The vector $v(t_k)$ represents the uncorrelated measurement error.

$$\begin{aligned} q(t_{k+1}) &= g(t_k)q(t_k) + w(t_k), k = 0,1,2... \\ z(t_k) &= h_k(q(t_k)) + v(t_k) \end{aligned} \qquad (2)$$

Furthermore, at each frame, an active selection among the three cameras allows the choosing of views which present the best information in order to deal with occlusions. The system is tested in a controlled environment which has uniform colour with no objects. This approach would be problematic if applied to unconstrained environments because of the complex background and clothing.

Deutscher et al. [32, 33] used three cameras and a modified particle filter on a simulated annealing algorithm to track a person. Compared to the standard condensation algorithm, the annealed particle filter (APF) reduces the number of samples and increases efficiency by a factor of up to 10. APF also localizes the tracked object even better as it increases the chances of finding the global minimum. Their body model is represented

by truncated cones and the cost function takes into account edge and silhouette information.

Carranza et al. [34, 35] corresponded an articulated body model with 2D data by overlapping between the observed 2D shapes and the projections of the model. The tracking was carried out by using Powell's method and via hardware acceleration. They achieved accurate results at about two seconds per frame on distributed hardware.

The methods presented above all exploit 2D image information for tracking. These cues only offer weak support to the tracker, which leads to sophisticated and slow optimization schemes. Multiple calibrated cameras allow the computation of the 3D shape of the person. The increase in the computational power offered by computers allowed real-time computation of the 3D shape and created several interesting approaches to full body tracking.

Cheung [36, 37] introduced the SPOT algorithm, a fast voxel-based method for the volumetric reconstruction of a person. Real-time tracking is achieved by mapping the voxels in the current frame to the closest body part of the previous frame. The position of the body-parts is updated at each frame. If registration is wrong, tracking of the body parts can be lost. Cheung later used both colour information and a shape-from-silhouette method for full body tracking, but the resulting system is not in real-time. Coloured surface points (CSPs) are used to segment the hull into rigid moving body parts, according to the results of the previous frames, and the constraint of equal motion of parts at their coupling joints to estimate joint positions. Cheung's system however, requires a complex initialization sequence to recover the joint positions of a person, which is used to track the same person in the following video sequences.

Mikic [38, 39] also proposed a voxel-based method for full body tracking. After volumetric reconstruction, sequential template growing and fitting is used to locate body parts after volumetric reconstruction. The fitting step uses the placement of the torso computed by the template growing to obtain a better starting point for the voxel labelling. An extended Kalman filter is used to estimate the parameters of the model given the measurements. To achieve robust tracking their method used prior knowledge of anthropometric measurements (average body part shapes and dimensions).

Mitchelson and Hilton [40] used shape and colour information for tracking the full human body. In their model-based approach they used a silhouette-overlap term to overcome the need for a volumetric reconstruction. For model initialization, a volume carving method is applied to retrieve the actual shape of the person's torso from a set of initialization images. An edge-proximity term and colour-consistency between the model and the images are needed to strengthen the fitness function for their hierarchical stochastic sampling scheme.

The major limitation of 3D based algorithms is that they often need initializing steps for mapping the colour and contour of the body and there is low accuracy of the reconstruction which mostly depends on the quality of the input images and the foreground segmentation. Furthermore, localization of the body parts is often necessary because voxel-based procedures tend to result in bulky reconstructions. Adding 2D cues can increase the tracking accuracy as they offer better localization. Moreover, the robustness against erroneous 3D reconstructions can be increased at the same time. Therefore, it may be best to combine 2D and 3D cues, so as to get the best of both worlds.

Plankers an Fua [41] achieved robust frontal, upper body tracking in the presence of self-occlusions by combining silhouettes and the depth information provided by three cameras. Body parts of the articulated body model are built from *metaballs* which offer realistic physical deformations. Tracking results are reported with self-occlusions of the arms against a controlled dark background.

The approach used in this research is markerless tracking with a single stereo camera. In comparison to such multiple camera systems in [29, 41] [30] [32], which make use of multi-views provided by a number of cameras; our proposed system works with a simpler hardware setup of only a single stereo camera, and uses three orthogonal silhouette views generated from the 3D stereo depth data. Our proposed system does not need initializing steps for colour and contour mapping and is robust against illumination, background and clothing variations.

2.4 Skin Colour Detection for Public Environments

In the last decade skin colour detection has become an often used cue in computer vision for detecting, segmenting, and tracking faces and hands [42, 43]. A complete list of 11 different colour spaces and their formulas are given in appendix A.

In this research, the skin colour based detection method was taken as the first approach to track the human head and arms. RGB values of each pixel were converted into HSL (Hue, Saturation, and Lightness or Luminance) colour space using equations 3, 4 and 5 [44]. Since in HSL space, brightness is stored in the element L, by discarding the L element, we can ignore the illumination information from video images; as a

result, the application is expected to be more resistant to illumination changes in the scene. H and S of each pixel are then compared to a skin-colour threshold (equation 6, 7) [45].

$$H = \cos^{-1} \frac{0.5((R-G)+(R-B))}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}} \tag{3}$$

$$S = \begin{cases} \dfrac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B) + \min(R,G,B)}, L \le 0.5 \\ \dfrac{\max(R,G,B) - \min(R,G,B)}{2 - (\max(R,G,B) + \min(R,G,B))}, L > 0.5 \end{cases} \tag{4}$$

$$L = \frac{\max(R,G,B) + \min(R,G,B)}{2} \tag{5}$$

$$0.23 \le saturation \le 0.63 \tag{6}$$

$$5° \le hue \le 90° \tag{7}$$

Despite the lightness component 'L' being discarded, we found this initial approach was still too sensitive to scene illumination levels because the same object illuminated by different lighting could have a completely different colour. This is even more problematic in a public interactive environment where a front projection method for the display is normally used. This means that colour lights from the projector are likely to shine on the user's body and reflect from surfaces in the scene, resulting in a complicated illumination. In addition, background and clothing with similar hues to those bounded by the skin manifold would also be identified as a human body part in this process. Figure 4 shows the result of using skin manifold detection in a poor lighting and background environment. From the bottom left picture, skin manifold

detection mistakenly picked up skin colour from the background owing to insufficient lighting, but omitted the hand and part of the facial skin owing to glare.

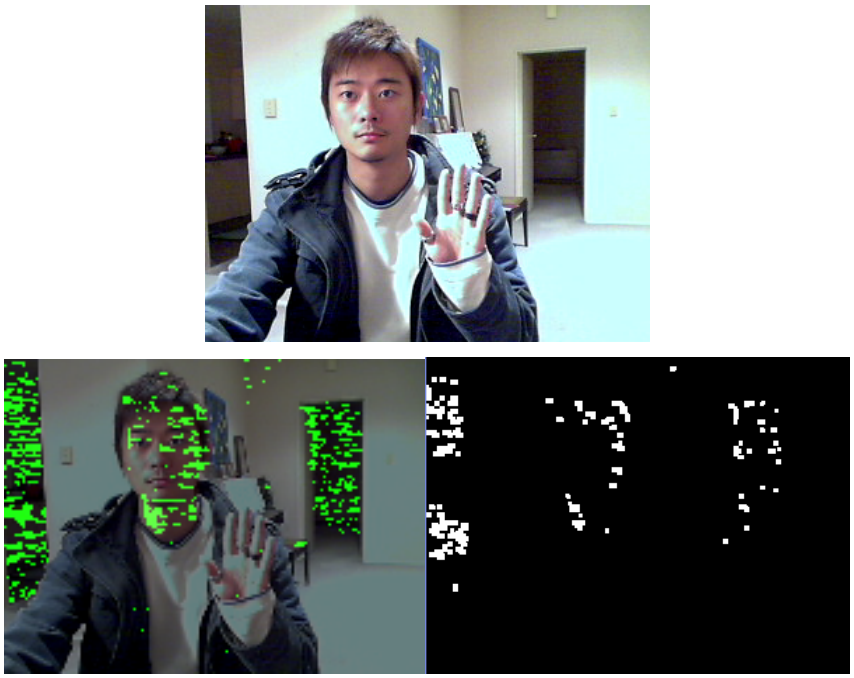


Figure 4. Performance of skin colour detection. Top: original image (under bad lighting). Bottom left: detected skin colour shown in green. Bottom right: processed binary image of the middle image (after noise removal).

In a public space, illumination can vary over time and even a small change in lighting has the potential to move the hue of a valid skin pixel outside the skin manifold. It is also problematic to constrain what people are wearing in a public space. For example, on a warm day, people may wear short sleeves or a singlet revealing large skin areas which are not covered, and so violate skin region constraints of only hands and face being visible. Conversely gloves may be worn on a cold day with the consequence that skin region tracking would fail to pick up the hands

simply because they are covered. Moreover, in the case of using front or floor projection techniques for interactive displays, coloured light is likely to be projected or reflected on to the user's body, to alter the skin colour seen by the camera.

In short, detecting body parts using regions of skin hue manifold is not favoured in public spaces owing to unconstrained conditions. Hence the motivation to use stereo camera systems is explained in the following section.

2.5 Stereo Tracking Survey

In the case of tracking head and arms in public environments, a markerless and non-colour cue based approach would be the most appropriate. The solution proposed in this thesis is to use a stereo camera system. The advantage of using such system is that it provides two adjacent views and 3D depth information of the scene. Therefore instead of segmented foreground / background based on colour cues, segmentation can be achieved based on the depth information. The following systems use stereo systems to track head and hands. Many have used colour cue to aid segmentation. Note that these tracking algorithms would be erroneous under variable illumination.

Azarbayejani and Pentland [46] presented a 3D head and hands tracking system which does automatic calibrations from watching a person. The algorithm is based on that of Darrel et al. [47] which uses colour, dense stereo processing and face pattern detection to track the head and silhouette.

Colombo developed the PointAt System [48] which allows users to walk around freely in a room within a museum, while pointing to specific parts

of a painting with their hand. Two cameras are set up to detect the presence of a person by using a modified background subtraction algorithm as well as skin colour detection. The tip of the pointing hand and the centre of the head are then extracted. By using visual geometry and stereo information, a pointing line is then deduced.

Nickel and Stiefelhagen [49] used a set of stereo cameras to track the user's hand and head to estimate the pointing direction in 3D space. The pointing gesture is recognised by using a three-phase model: *Begin* (the hand moves from an arbitrary position towards a pointing position), *Hold* (the hand remains motionless while pointing) and *End* (the hand moves away from a pointing position). Comparing three approaches to estimate pointing direction, they found that the hand-head line method was the most reliable in estimating the pointing direction (90%). Owing to the reported high accuracy, this algorithm for pointing gestures is employed in this proposed upper limb tracking system.

Javier and Jose [50] presented a robust real-time 3D tracking system of human hands and face for VR applications. The system employs a stereo pair, and the algorithm is based on a skin colour module and a hypothesis based data association algorithm to track the user's hands and face. The application also outputs the 3D reconstruction of the user to allow a human-computer interaction system for a virtual reality environment.

Jun, and Joongeon [51] proposed a new real-time system to acquire motion information of human articulated objects such as arm and head. The motivation for using a stereo camera was to achieve markerless detection as well as resistance to illumination change and complex background. The idea of the proposed system is to apply component labelling techniques on a sliced disparity map to locate arm position.

Munoz-Salinas and Garcia-Silvente [52] presented an approach for multiple-people detection and tracking using stereo vision. Tracking is carried out using a multiple particle filtering approach that combines depth, colour and gradient information. The proposed algorithm makes use of the information available (colour and gradient) to track and combines adaboost classifiers with stereo information. Their results show that the algorithm is able to deal with occlusions and effectively determine both 3D and 2D head locations in the camera images.

Gordon, Cheng and Buck [53] proposed an algorithm for a person and gesture tracking system based on the TYZX smart stereo cameras. They suggested that using stereo cameras provides tracking accuracy in dynamically lit environments and such 3D imaging technology is not affected by constant changes in lighting and apparent colour. In addition stereo cameras can provide the location and movement of each individual very precisely.

2.5.1 SRI SVS and Point Grey Stereo Cameras

In the early stage of this research, the performances from two different stereo cameras are compared. They are the Videre Design's SRI Small Vision Systems (SVS) stereo cameras [1] (Figure 5a) and Point Grey Research's Digiclops stereo system[2] model Bumblebee2 (Figure 5b).

[1] www.videredeign.com
[2] http://www.ptgrey.com/products/stereo.asp

(a) (b)

Figure 5. (a) SVS Stereo Camera System.
(b) Pt. Grey Research's Digiclops Bumblebee2 Camera System

According to Urmson, C [54], Point Grey's Digiclops system is less robust in terms of speed (or throughput). Table 1 shows the comparison between the two systems.

	SVS with IEEE1394 Cameras (ms) (over 100 samples)	Digiclops (ms) (over 20 samples)
Image capture	104.5	122.8
Warp	10.7	89.4
Stereo	48.6	305.3
Whole Cycle	**163.8**	**517.5**

Table 1. Comparison of the time performance of SVS and Digiclops.

Despite SVS being faster, the bumblebee stereo camera was chosen to acquire image and depth information for this research. The reasons for using the Bumblebee2 stereo camera over the SVS stereo camera are that Bumblebee2 cameras come pre-calibrated and provide higher image resolution than SVS systems. In terms of the API of stereo systems, Digiclops uses Triclops SDK which provides more ability to customize the behaviour of the stereo engine than SVS systems. It also provides

more features, for example, surface validation, disparity scaling and strict sub pixel validation. In addition, The Digiclops system produces better output. In disparity images generated by the Digiclops, obstacles appeared sharper and data seemed to be more reliable at slightly longer ranges than those available from the SVS system.

2.6 2D and 3D Models for Whole Body Tracking

Various approaches for tracking the whole body have been proposed in the literature as listed in Table 2. They can be distinguished according to their representation of the body as a stick figure, 2D contour or volumetric model and by their dimensionality being 2D or 3D.

As presented in the literature [55, 56], one way to recover 3-D body pose is to work directly with 2-D features derived from the images, using some form of 2-D model. Recognition systems that use 2-D model-free features have been able to claim successes in matching human movement patterns. In [57], constrained types of human motions such as walking parallel to the camera image plane and periodic motion, have been successfully used for classification. This may be the easiest and best solution for several applications, but it is unlikely to achieve reliable recognition for more unconstrained and complex human movement such as in a public interactive environment. Using these types of features exclusively, recognition would be error prone when for example, a person making different gestures while walking around and turning. In addition, self-occlusion makes the 2-D tracking problem even harder for arbitrary movements; therefore most of the existing systems assume some prior knowledge of the type of movement and/or the viewpoint under which it is observed. 2-D labelling and tracking under more general conditions are attempted by [56].

	Model Based Tracking			Feature Based Tracking
	Stick Model	2D Contour Model	3D Volumetric Model	
(Rogez et al., 2007)[58]	●			
(Karaulova et al., 2000)[59]	●			
(Wren et al., 2000)[60]	●			
(Iwai et al., 1999)[61]	●			
(Yaniz et al., 1998)[62]	●			
(Schrotter G. et al, 2005)[63]	●			
(Remondino F. 2003)[64]	●			
(Theobalt C. et al., 2002)[65]	●			
(Moreno-Noguer et al., 2007)[66]		●		
(Panin et al., 2006) [67]		●		
(Leung and Yang, 1995)[56]		●		
(Black and Yaccob, 1996)[68]		●		
(Kameda et al., 1995)[69]		●		
(Hu et al., 2000)[70]		●		
(Yuan et al., 2007)[71]			●	
(Urtasun et al, 2006)[72]			●	
(Kehl et al., 2005)[73]			●	
(Wachter and Nagel, 1999)[28]			●	
(Delamarre, 1999)[74]			●	
(Urtasun R. and Fua P. 2004)[75]			●	
(Luck et al., 2001)[76]			●	
(Ladikos et al., 2007)[77]				●
(Jang and Choi, 2000)[78]				●
(Rosales and Sclaroff, 2000)[79]				●
(Krinidis M. et al., 2005)[80]				●
(Nguyen et al., 2001)[81]				●

Table 2. Comparison of different human body models.

The stick figure is simply a collection of segments and joint angles with various degrees of freedom at the articulation joints. Volumetric 3D models have the ability to resolve self occlusions easily [82] and they also allow 3D joint angles to be directly estimated by mapping 3D body models on to a given 2D image. Most volumetric approaches model body

parts using generalized cylinders [83] or super-quadratics [84]. Some extract features [85], others fit the projected model directly to the image[83].

2.7 Interaction Study

2.7.1 Gesture Based Interactions

Since the 1960s and 70s, many have admired Larry Tesler (who introduced cut and paste), Doug Engelbart (selecting, point and click, windows), and Tim Mott (the desktop metaphor) for their modern set of interaction paradigms that we have used ever since. Now we need to take into consideration that the current robust processing power in modern computers poses new opportunities for new interaction paradigms using arm and hand gestures. Individual companies (Apple, Nintendo etc) have recently created and introduced their own physical interactions embedded in their product (Iphone, Ipod Touch, Wii) to the public. However because it would require remembering a large number of movements and gestures for the same common actions, it is necessary to propose a common set of movements and motions that could be used for initiating actions across a variety of platform devices and environments.

This section presents the taxonomy (figure 6) of interactive gestures employed in the current devices. In particular, we focus on taxonomy by gesture styles. Direct manipulative gestures are discussed in detail as this is related to the interaction part of the research (Chapter 4). A complete detailed description of the taxonomies of interactive gestures could be acquired in [86].

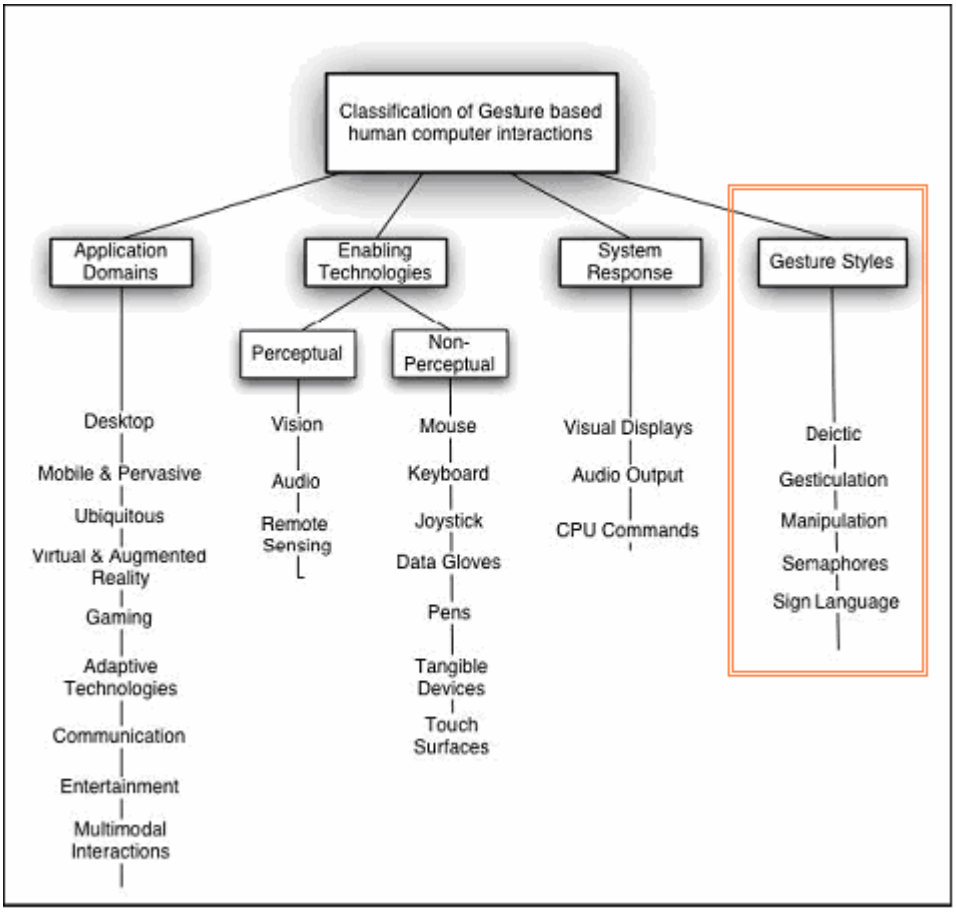


Figure 6. Visualizing the taxonomy: The diagram shows the organization of the research on gestures based on the four categories used in the taxonomy. In this section, taxonomy by gesture styles is explained in detail.

2.7.2 Deictic Gestures

The most basic and intuitive gesture is pointing, also known as deictic gestures. They involve pointing to establish the identity or spatial location of an object. This is an effective method for most people to communicate with each other, even in the presence of language barriers. However, pointing as a way to communicate fails when the concept domain that is to be conveyed is too complex or not in sight to point at. Deictic gestures are often considered to be embedded in other forms of

gestures, such as when pointing to identify an object to manipulate for example [87, 88].

2.7.3 Manipulative Gestures

According to Quek [89] the primary purpose of manipulative gestures is to control some entity by applying a tight relationship between the actual movements of the gesturing hand/arm with the entity being manipulated. Manipulations can occur both on the desktop in a 2-dimensional interaction using a direct manipulation device such as a mouse or stylus, as a 3-dimensional interaction involving empty handed movements to mimic manipulations of physical objects as in virtual reality interfaces, or by manipulating actual physical objects that map onto a virtual object in tangible interfaces.

2.7.4 Gesturing in Two Degrees of Freedom for 2-Dimensional Interactions

Many human computer interactions often involve manipulation of 2-dimensional objects on the display such as a cursor or a window. The traditional interaction method for such 2 dimensional objects normally involves the mouse, stylus or other 2 degrees of freedom (DOF) direct input devices for direct manipulation of objects in a graphical user interface (GUI). Direct manipulations are interactions associated with desktop applications and consist of actions such as dragging, moving and clicking objects. The current common gestures for direct manipulation are shown in figure 7 and listed in table 3. In this research, a set of interaction gestures for large interactive displays is proposed based on the current direct manipulation inputs in iPhone and iPod touch. A complete set of direct manipulation is presented in chapter 4.

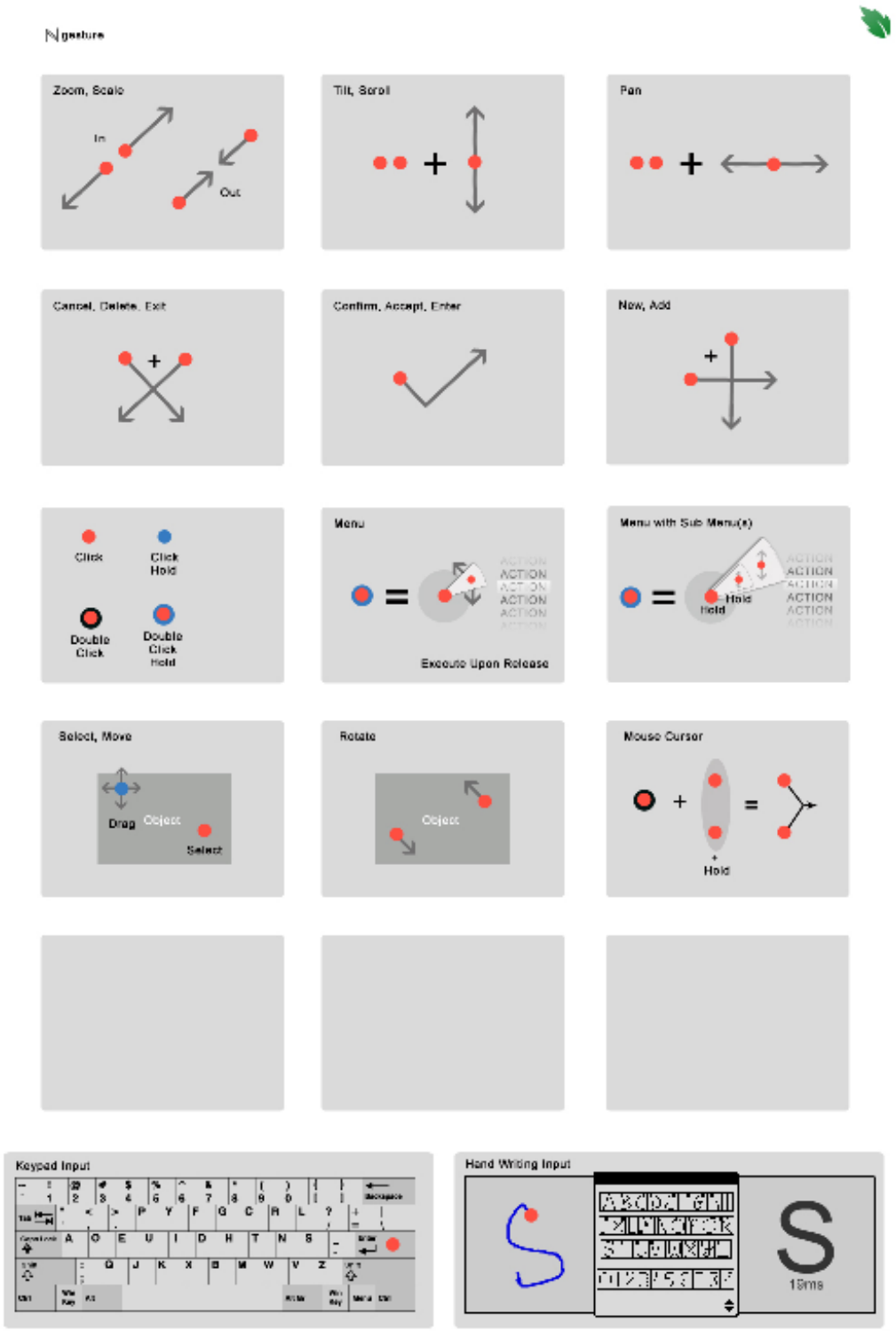


Figure 7. Visuals of common direct manipulation gestures.

Gestures for Direct Manipulation	Description
Circular scroll	Slides a finger around in a circular motion to scroll forward/down/right. A clockwise gesture scrolls down/forwards/right, counter-clockwise scrolls up/backwards/left.
Flick to move left/right	Quickly sliding a finger (usually index) to the left or right in a flicking motion moves to the next item in a scrolling list.
Flick to momentum-scroll up/down	Quickly sliding a finger up or down in a flicking motion causes the content of the screen to scroll in that direction. Based on the speed of the flick, the list will continue scrolling after the gesture is complete, slowing to a gentle stop.
Pinch to shrink	An on-screen object is "grasped" between two fingers, typically between thumb and index finger. As the user moves his fingers together, the object decreases proportionately in size.
Point to select	A long single tap. This calls any secondary functions available on the tapped screen element. This is equivalent to the right mouse click in desktop OSs.
Slide finger to move object	Known as Tap & Drag. Equivalent to Onmousedown & Onmousemove in desktop OS.
Spread to enlarge	An on-screen object is touched by two fingers. As the user moves the fingers in opposite directions (spreading them apart), the selected object grows in size.
Two finger slide to scroll	Two fingers when pressed together and touched on the screen can scroll the screen up/down or left/right by sliding together in a direction.
Tap to open/activate	Known as "tap," Equivalent to a left button mouse click in desktop OSs.
Tap to stop momentum-scrolling	During a momentum-scroll, pressing a finger against the screen will immediately stop the scrolling.
Torque to rotate	Two fingers touched on the screen and rotate clockwise or anti-clockwise.

Table 3. List of direct manipulative gestures.

2.7.5 Gesturing in Multiple Degrees of Freedom for 2-Dimensional Interactions

3-dimensional (3D) style manipulations of 2D objects normally involve gestures such as physically picking up and dropping a data file from one device and virtually moving it to another device usually located within a smart room environment [90]. In recent work, table top surfaces are fitted with electronic material designed to sense pressure and touch as well as movement and multiple points of finger and hand contact. Several systems including Rekimoto's smart skin enable manipulative gestures that are drawn from actual table top interactions such as sweeping and isolating groups of objects with the hands as one would do with physical objects on a table for example [91, 92].

2.7.6 Semaphoric Gestures

Semaphores[3] are systems of signalling using flags, lights or arms. By extension, semaphoric gestures are defined as any gesturing system that employs a set of static or dynamic hand or arm gestures. Semaphoric approaches may be referred to as "communicative" since the gestures serve as a universe of symbols to be communicated to the machine. Semaphoric gestures are also one of the most widely applied styles, even though the concept of using signs or signals to communicate information has been a small part of human interactions [89], and provides little functional utility [93]. However, with the movement towards more ubiquitous computing paradigms, the use of semaphoric gestures is seen as a practical method of providing distance computing in smart rooms and intelligent environments. There are several different forms of

[3] Brittanica.com

gestures that fall into the category of semaphores discussed in the literature which will be described next.

Semaphoric gestures can involve static poses or dynamic movements unlike manipulative gestures which are mainly dynamic. For example, when the thumb and forefinger are joined to represent the “ok” symbol, this is a static pose, while moving the hand in a waving motion is a dynamic semaphoric gesture. These types of gestures can be performed using a hand [92, 94, 95] fingers [96, 97] arms [49, 98] the head [99, 100] feet [101] or other objects such as passive or electronic devices such as a wand or a mouse [95, 102, 103].

2.7.7 Language Gestures

Gestures used for sign languages are often considered independently of other gesture styles since they are linguistically based and are performed using a series of individual signs or gestures that combine to form grammatical structures for conversational style interfaces. An example would be the finger spelling (figure 8), and sign languages can be considered semaphoric in nature. However the gestures in sign languages are based on their linguistic components, and although they are communicative in nature they differ from gesticulation in that the gestures correspond to symbols stored in the recognition system.

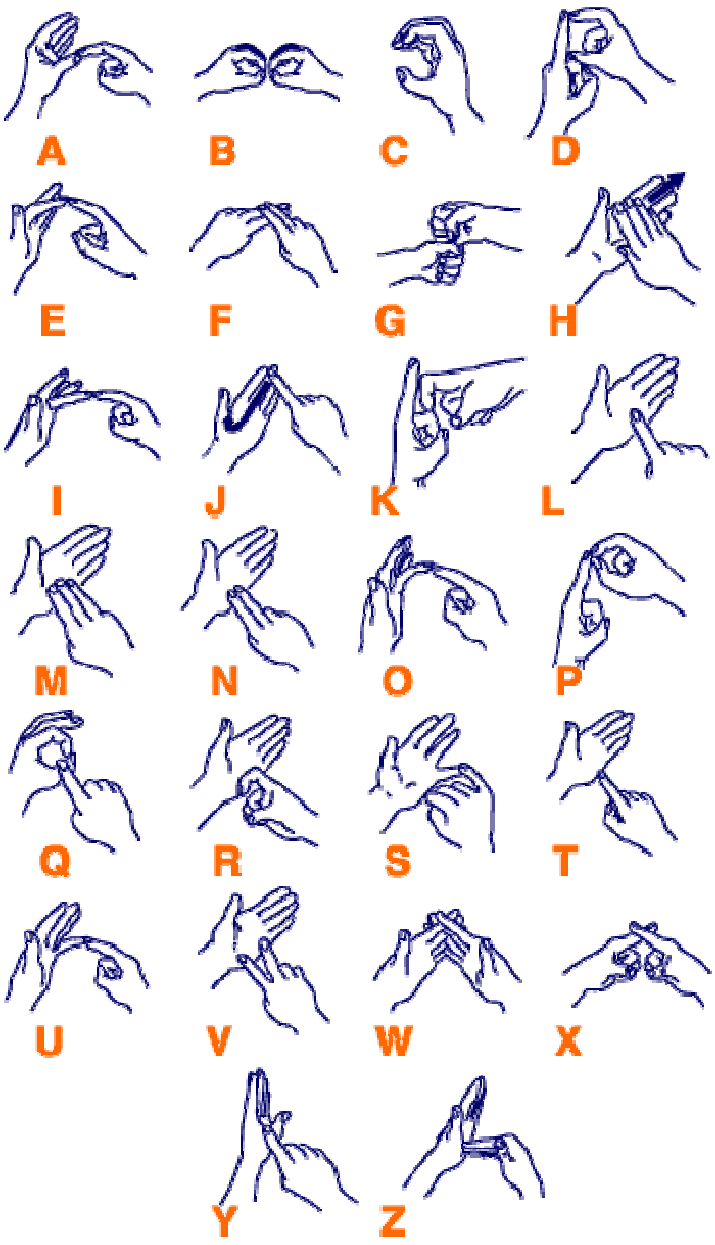


Figure 8. Alphabet of New Zealand finger spelling.

2.8 Summary

In this chapter, computer vision systems are discussed briefly with specific focus on markerless detection for the human head and arms. According to the discussion on colour based detection techniques in section 2.4, we conclude that such detection techniques are inadequate for tracking the human face and arms in a public environment where illumination, clothing and background are varying factors. Stereo systems are then introduced as the alternative for robust tracking in such environments. The literature on model based tracking is provided in section 2.6. The taxonomy of different types of gestures is also introduced. Amongst these, the focus lies on direct manipulative gestures as this thesis later proposes a set of interactions for large interactive displays based on these gestures in Chapter 4.

3. Proposed Upper Limb Detection Algorithm

3.1 Introduction

This chapter describes the proposed system which recognizes arm gestures and tracks upper limb motions. As discussed in the previous chapter, tracking and recognizing body parts using a colour manifold can often produce undesirable results due to variable illumination, clothing and background. Unfortunately this is often the case in a public environment. To overcome these shortcomings, this thesis proposes a system which is based on a single stereo camera.

Recognizing human faces is seen as one of the important components of the proposed system. The main purpose for face tracking is that the position of the face is used in the process of forearm segmentation. Furthermore, the position is also exploited when finding the eye-hand vector which improves pointing in the user's perception.

This thesis proposes a new algorithm which interprets 3D stereo data into a more comprehensible set of three 2D images. This is equivalent to having three separate orthogonal cameras, which provides more information to help understand the observed scene. A detailed discussion is provided in section 4.7.

3.2 Video Acquisition

The first step in any computer vision application is to acquire a digital video of the problem domain. This is normally done by having digital cameras connected to a computer. Unlike film-based cameras, digital cameras have an image sensor that converts light into electrical charges

[104]. The image sensor employed by most digital cameras and web cams is a charge coupled device (CCD). Some low-end cameras use complementary metal oxide semiconductor (CMOS) technology which improves the image quality, however it is comparatively slower than CCD cameras.

The amount of detail that the camera can capture is the resolution, and it is measured in pixels. The bigger the resolution the camera gives the more detail it can provide. Higher resolution video frames can enhance the performance of pre-processing and segmentation algorithms in computer vision applications which lead to higher accuracy. Nevertheless, higher resolution can contribute to increased processing power and time.

In computer vision, systems use either a single camera (monocular vision) or multiple cameras (multi-view) which depends on the problem domain. Either way, an accurate camera calibration is always a requirement.

3.3 Camera Calibration

Camera calibration in the context of computer vision is the process of determining the intrinsic and extrinsic parameters of the camera [24]. The intrinsic parameters include the internal geometry of a camera such as camera constant, the location of the principal point and corrections for lens distortions. Extrinsic parameters include position and orientation of the camera in an absolute coordinate system from the projections of calibration points in the scene.

The overall performance of the computer vision system strongly depends on the accuracy of the camera calibration. Camera projection is often modelled with a simple pinhole camera model. Several methods for camera calibration are presented in the literature. The classic approach

[28] that originates from the field of photogrammetry solves the problem by minimizing a nonlinear error function. Owing to the slowness and the computational burden of this technique, closed-form solutions have been also suggested (e.g. [24], [25], [26]). However, these methods are based on certain simplifications in the camera model, and therefore they do not provide such good results as nonlinear minimization. There are also calibration procedures where both nonlinear minimization and a closed form solution are used. In these two step methods, the initial parameter values are computed linearly and the final values are obtained with nonlinear minimization. The methods where the camera model is based on physical parameters, such as focal length and principal point, are called explicit methods. In implicit camera calibration, the physical parameters are replaced by a set of non-physical implicit parameters that are used to interpolate between some known tie-points (e.g. [27]).

A general strategy for calibration is to view a calibration target such as a checkerboard pattern (figure 9), identify the image points and obtain a matrix called the camera matrix using one of the techniques above.

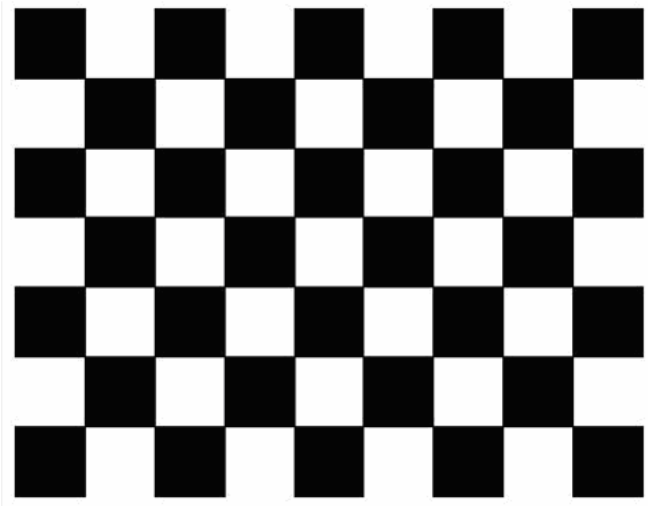


Figure 9: A Calibration Target

3.4 Monocular Vision

In monocular vision, only a single camera is used to obtain a video of the real world. Normally it is used in applications where the depth of the objects in the scene is not required. There are no added geometrical constraints to the camera as is the case with stereo vision. However, camera calibration is still essential.

3.5 Stereo Vision

Depth information at each pixel can be a useful cue for efficient background subtraction and 3-D reconstruction of the scene. 3-D information can be estimated indirectly from 2-D intensity images using image cues such as shading and texture [105]. Both shading and texture are considered to be indirect methods and are not accurate. In order to compute real depth at each pixel, stereo vision is used.

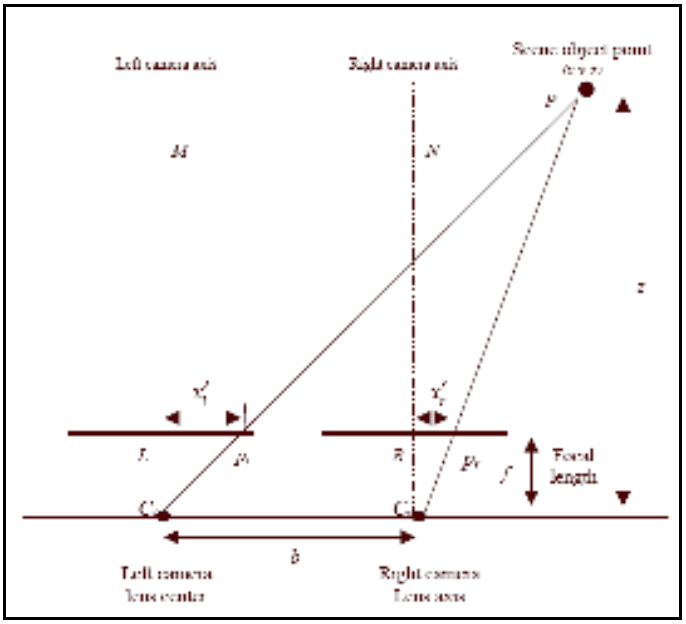


Figure 10. Epipolar geometry. Any point in the scene that is visible in both cameras will be projected to a pair of image points in the two images, called a *conjugate pair*. The displacement between the positions of the two points is called the *disparity*.

The epipolar geometry or epipolar constraint for stereo vision is shown in figure 10. In the simplest model, two identical cameras are separated only in x direction by a baseline distance b. In this model, the image planes are coplanar. A feature in the scene is viewed by two cameras at different positions in the image plane. The displacement between the locations of the two features in the image plane is called the disparity. The plane passing through the camera centres and feature point in the scene is called the epipolar plane. The intersection of the epipolar plane with the image plane defines the epipolar line. A conjugate pair is two points in different images that are the projections of the same point in the scene. In figure 10, the scene point P is observed at points Pl and Pr in the left and right images, respectively. If by assumption, the origin of the coordinate system coincides with the left lens centre, then by comparing the similar triangles P-M-Cl and Pl-L-Cl:

$$\frac{x}{z} = \frac{Xl}{f} \tag{8}$$

Similarly from the similar triangles P-N-Cr and Pr-R-Cr:

$$\frac{x-b}{z} = \frac{Xr}{f} \tag{9}$$

Combining these two equations:

$$z = \frac{bf}{(Xl - Xr)} \tag{10}$$

Thus the depth at various points in the scene may be recovered by knowing the disparities of corresponding image points. Equations for

depth values for cameras in arbitrary positions and orientation are more complex and are discussed in [106].

The above mentioned technique is based on the assumption that conjugate pairs in stereo images can be identified. Detecting conjugate pairs has been an extremely challenging research problem known as the correspondence problem. In order to solve this correspondence problem, it is necessary to find every point from the left image and correspond it with the point on the right image. The epipolar constraint significantly limits the search space for finding conjugate pairs.

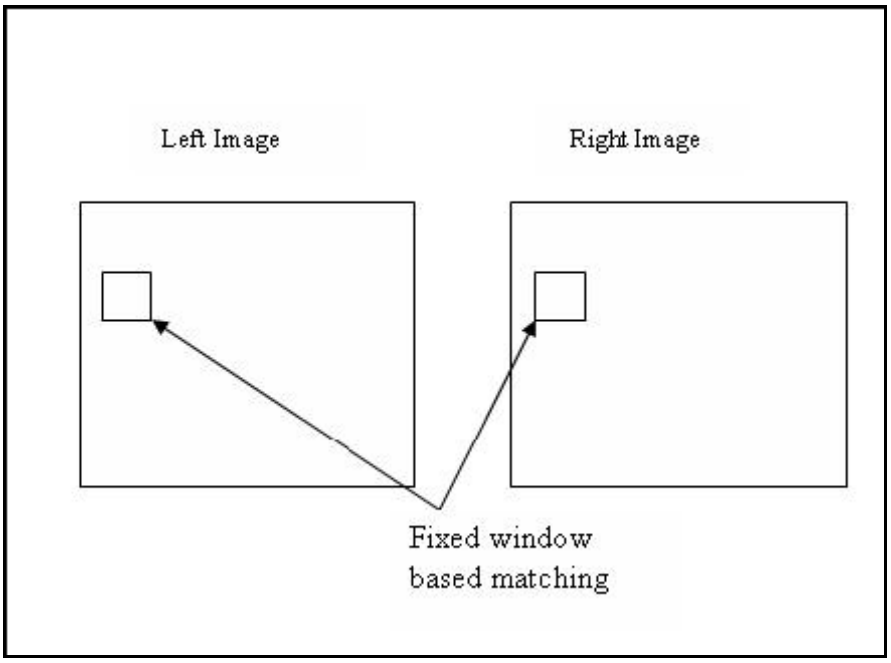


Figure 11: Stereo Matching using Fixed Size Window

Stereo correspondence is an area where extensive research has been carried out. Stereo algorithms can be classified either as local (window matching as in figure 11) or global. In window-based stereo matching or local matching, the problem is to identify an optimal support window for each pixel. An ideal window region should be bigger in non-texture

regions and should be suspended at depth discontinuities. Fixed window based approaches are invalid at depth discontinuities. Some improved methods, such as adaptive windows [106], shiftable windows [107] and compact windows[108] try to avoid the problems at depth discontinuities. Bayesian methods [109-111] are global methods that try to model discontinuities and occlusion using statistical techniques.

As discussed above, stereo vision can help determine depth information of the real world. This information can be used for background subtraction and many other segmentation techniques. Much of the ambiguity in the scene could be removed by identifying pixels close to cameras. For example, if a motion capture application is placed in a public space, it would be desirable to identify a human who moves in front of the camera/cameras from a human walking in the background. In addition, disparity values obtained from stereo cameras are less sensitive to lighting. Monocular vision would not distinguish how far away an object is as it does not give us any depth information of the scene.

3.6 Feature Based Face Detector

In this research, a face detector is used to recognize any human faces in the video frame and it activates the system to begin recognition and further tracking processes.

Research into face detectors takes fundamentally different approaches. Selecting the right detector is an important step in the design of pose estimation systems, because the quality of the output affects the overall system performance. The extent to which a system's performance is affected by the face detector varies, and because speed is often an issue in addition to accuracy for these systems, selecting the right face detector must be done with care.

3.6.1 Face Detectors

This section gives an overview of some general face detection approaches with examples:

a) Low-level features

This class of face detectors works by extracting a set of low-level features from a region, followed by making the binary classification of whether the region is face or non-face. Feature based face detection normally works in the form of a boosted classification scheme of simple features easily calculated as in [112]. This approach uses a cascade of simple classifiers boosted by adaptive boosting, or AdaBoost, as introduced by [113]. The features used are insensitive to illumination changes and also invariant towards person identity.

b) Neural networks

Regions of a fixed size are fed to a neural network which output some likelihood that the region is a face. The neural network approach by Rowley[114] is based on texture analysis. Feraud [115] also uses motion features and colour information as input to the neural net. Before the introduction of the AdaBoost face detector, the neural network approaches were considered the most robust as well as the fastest feature based face detectors [112].

c) Colour cue

There are many tracking systems which use colour based skin segmentation approaches and have reported high performance rates [116]. The colour based face detectors all work in similar ways. Large, skin coloured, head-shaped blobs are identified as faces. The main difference between different skin colour face detector algorithms lies in the way a

region is determined to be skin coloured or not and how to constrain the head shape. Compared to the human visual system, the colour based approaches are more sensitive to illumination changes. For humans it is basically the direction of the illumination that causes problems, whereas colour based approaches also suffer from illumination changes in situations in which the colour temperature changes, such as when moving from direct sunlight to shade or moving between indoors and outdoors.

d) Top-down models

In the top down models, face detection is performed using template matching of the entire face. In [117], an image is searched first with a coarse template and then with templates with finer and finer detail. The top-down approaches excel in images with multiple faces.

e) Bottom-up models

Once feature candidates have been detected, their distance relationship or relative position is used to fit a bounding box around [118, 119]. For robust performance the bottom-up approach is limited to single face images, owing to the difficulty of accurately distinguishing multiple feature candidates from the multitude of false detections under varying conditions. It is easier to select the best match from a set of candidates than to set a threshold for keeping a manifold of candidates constant.

3.6.2 Recognizing Faces using Haar Classifiers

When selecting among feature based face detectors for providing well aligned face regions, the face detector of Viola and Jones [112] is employed in many computer vision systems because of the high performance rate it provides. The proposed approach is promising and the detector has a number of appealing qualities: it is extremely fast, and the speed is achieved without compromising the need for invariance

properties and robustness. It is accessible as the Haar-face from the OpenCV library, therefore it was chosen for this research. In the proposed algorithm, it serves the purpose of triggering the detection process when a person's face is visible to the camera while providing resistance to illumination and background changes.

The core basis for Haar classifier object detection is the Haar-like features. These features, rather than using the intensity values of a pixel, exploit the change in contrast values between adjacent rectangular groups of pixels. The contrast variances between the pixel groups are used to determine relative light and dark areas. Two or three adjacent groups with a relative contrast variance form a Haar-like feature. Haar-like features, as shown in Figure 12 are used to detect an image. Haar features can easily be scaled by increasing or decreasing the size of the pixel group being examined which allows features to be used to detect objects of various sizes.

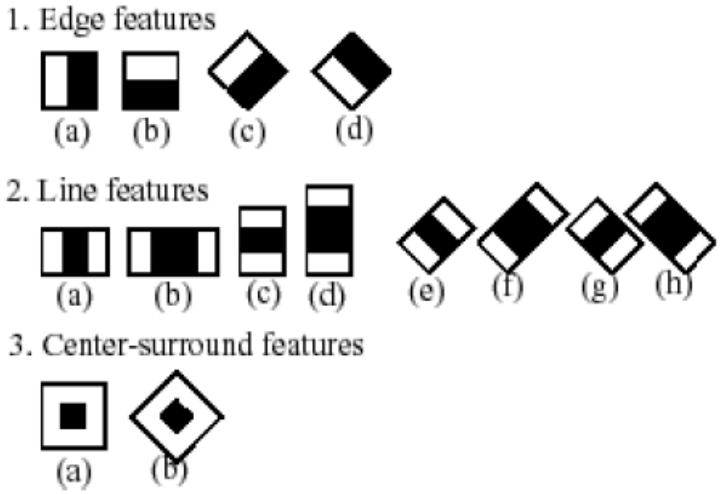


Figure 12. Common Haar Features

The features are used to extract information about the region fed to the detector. The feature values are calculated in the region, at all possible

locations and all imaginable scales in both the horizontal and vertical direction. An example of the application of the features is shown in Figure 13. The features shown in the depicted location at that scale are semantically meaningful. The horizontal feature at that location extracts information about the eyes and the vertical extracts nose bridge information.

Figure 13 Horizontal and vertical features. The features are calculated at a given location at a given scale in the region tested. The white rectangle indicates the region, and the features are placed somewhere within that region. In this case features extract information about the dark eye region and the highlighted nose bridge.

3.6.3 Integral Image

The simple rectangular features of an image are calculated using an intermediate representation of an image, called the integral image [112]. The integral image is an array containing the sums of the pixels' intensity values located directly to the left of a pixel and directly above the pixel at location (x, y) inclusive. So if A[x,y] is the original image and AI[x,y] is the integral image then the integral image is computed as shown in equation 11 and illustrated in Figure 14.

$$AI[x,y] = \sum_{x' \le x, y' \le y} A(x', y') \qquad (11)$$

The features rotated by forty-five degrees, require another intermediate representation called the rotated integral image or rotated sum auxiliary image [120]. The rotated integral image is calculated by finding the sum of the pixels' intensity values that are located at a forty five degree angle to the left and above for the x value and below for the y value. So if A[x,y] is the original image and AR[x,y] is the rotated integral image then the integral image is computed as shown in equation 12 and illustrated in Figure 15.

$$AR[x,y] = \sum_{x' \le x, x' \le x - |y - 'y|} A(x', y') \qquad (12)$$

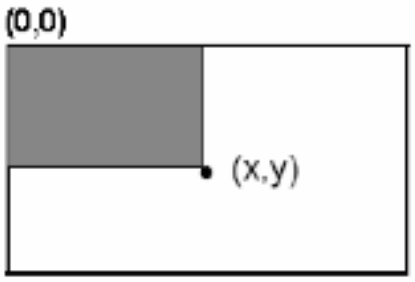


Figure 14. Summed area of integral image

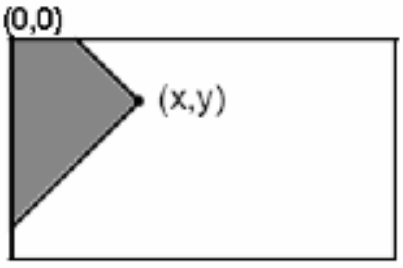


Figure 15. Summed area of rotated integral image

It only takes two passes to compute both integral image arrays, one for each array. Using the appropriate integral image and taking the difference between six to eight array elements forming two or three connected rectangles, a feature of any scale can be computed. Thus calculating a feature is extremely fast and efficient. It also means that calculating features of various sizes requires the same effort as a feature of only two or three pixels. The detection of various sizes of the same object requires the same amount of effort and time as objects of similar sizes since scaling requires no additional effort [112].

3.6.4 Classifiers Cascaded

Although calculating a feature is extremely efficient and fast, calculating all 180,000 features contained within a 24 × 24 sub-image is impractical [112]. Fortunately, only a tiny fraction of those features are needed to determine if a sub-image potentially contains the desired object [121]. In order to eliminate as many sub-images as possible, only a few of the features that define an object are used when analyzing sub-images. The goal is to eliminate a substantial amount, around 50%, of the sub-images that do not contain the object. As this process continues, more features are used to analyse the image at each stage. The cascading of the classifiers (figure 16) allows only the sub-images with the highest probability to be analyzed for all Haar-features that distinguish an object. It also allows one to vary the accuracy of a classifier. One can increase both the false alarm rate and positive hit rate by decreasing the number of stages. The inverse of this is also true. Viola and Jones were able to achieve a 95% accuracy rate for the detection of a human face using only 200 simple features [112]. Using a 2 GHz computer, a Haar classifier cascade could detect human faces at a rate of at least five frames per second [120].

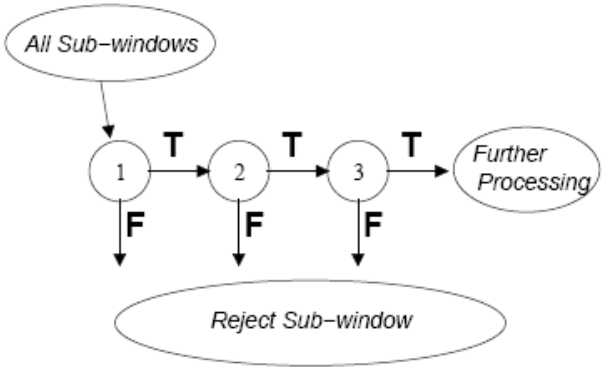


Figure 16: Schematic depiction of a detection cascade. A series of classifiers are applied to every sub-window. The initial classifier eliminates a large number of negative examples with very little processing. Subsequent layers eliminate additional negatives but require additional computation. After several stages of processing, the numbers of sub-windows have been reduced radically.

3.7 Anthropometric Constraints

According to da Vinci's Vitruvian[4] Man, the average adult human figure is about 7 to 7.5 heads tall, and the length of a human's outspread arm is roughly equal to the height. Although research has since dispelled da Vinci's myth of an average human, anthropometric data supports such approximations as useful.

Anthropometry refers to the measurement of living human individuals for the purposes of understanding human physical variation. Today, anthropometry plays an important role in industrial design, clothing design, ergonomics, and architecture where statistical data about the distribution of body dimensions in the population are used to optimize products.

[4] http://en.wikipedia.org/wiki/Vitruvian

The measurements are arranged in anthropometric data tables usually by sex and percentiles. There exist 5^{th}, 50^{th}, and 95^{th} percentile measurements. These refer to portions in statistic normal distribution. For example, the 50^{th} percentile height measurement is mean height in a group of people; the 5^{th} percentile measurement would only cover 5% of the population (1 in 20 people).

For this proposed research, a novel upper limb detection algorithm which would be robust for most potential users is proposed. Therefore, the 95th percentile measurement is chosen to target a good diversity of people. The Zone of Convenient Reach (ZCR) is also considered so that when users interact with the proposed system, any movements can be reached conveniently, that is without undue exertion. Consider what it means for a control to be 'within arm's length'. The upper limb, measured from the shoulder to the fingertip, sweeps out a series of arcs centered upon the joint (Figure 17). These define the zone of convenient reach for one hand, which extends sideways to the coronal plane of the body. The zones for the two limbs intersect in the midline (median) plane of the body. The volume which is thus defined comprises two intersection hemispheres. The radius of each hemisphere is the upper limb length (a) and their centres are a distance (b) equal to biacromial breadth apart.

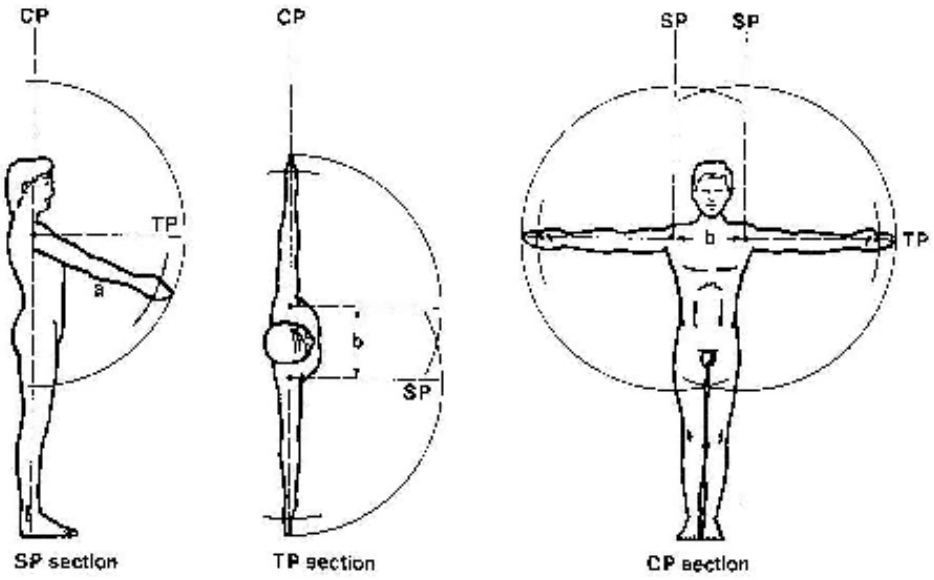


Figure 17. Zone of Convenient Reach (ZCR) seen in elevation and plan. (left to right) Vertical section in saggital plane (SP) passing through shoulder joint; horizontal section in transverse plane (TP) passing through shoulder joints; vertical section in coronal plane (CP) passing through shoulder joints. Each plane of the section is marked on the other two diagrams.

Table 4 shows Pheasant's 95th percentile proportions for both sexes, which have relevant measurements used to define anthropometric constraints in this research. The measurements are in millimetres. Refer to [122] for a complete list of anthropometric measurements.

The algorithm described by this thesis makes use of Pheasant's 95th percentile measurements for men, because according to the estimates in table 4, men generally have larger dimensions than women; therefore by using men's measurements, this would also cover the 95th percentile for women.

Dimension	Men's 95th %ile	Women's 95th %ile
Eye height	1745	1610
Shoulder height	1535	1405
Shoulder-elbow length	395	360
Elbow-fingertip length	510	460
Upper limb length	840	760
Head length	205	190
Head breadth	165	150
Shoulder breadth (biacromial)	430	385
Span	1925	1725

Table 4. Anthropometric estimates for the 95th percentile of British adults aged 19-64 years. All dimensions in millimetres.

The human arm rotates around the shoulder joint, which is about one face length (20.5 cm) below the centre of the face[123]. The shoulder breadth is 43cm wide. Therefore an interaction volume is defined for the individual users by two spheres centred at the height of the shoulder with diameters of upper limb length. This is similar to the ZCR in figure 17. However for ease of searching within the interaction volume, the two spheres have been merged into one big sphere, centred at the midpoint between shoulders with a diameter of arm span (192 cm).

It is reasonable to assume that, when interacting with the display, people tend to move their dominant hand towards the display and sufficiently away from the face (>30cm). Thus the hand search space is restricted to a volume delimited by a sphere bisected by the frontal plane (Figure 18) [124]. Therefore the hand search is limited to just a semi-spherical space with a radius up to 30 cm in front of the face and no further than 84 cm (upper limb length). Anything inside this semi-sphere is considered a

valid location for human hands/arms. Anything that falls outside is discarded. If a pose is recognized and is inside the volume, it is assumed that the user wants to interact with the system. This interaction space serves to prune the pose search space for efficient pose validation.

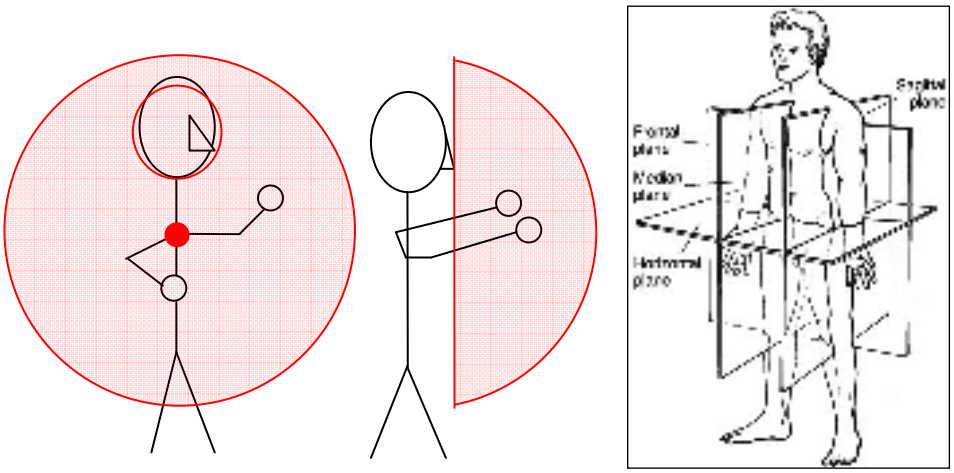


Figure 18. Left: frontal view of the user; small circle indicates the detected face, large circle indicates the interaction space. Centre: side view of the user, semi-sphere indicates the interaction space is 30 cm in front of the user. Right: Anatomical planes in a human.

3.8 Head and Hand Detection Algorithm

In order to trigger a hand search, a face must first be recognized in the current frame. A cascade of Haar classifiers[5] is used to scan the image several times at different scales applying Canny pruning reduces the number of analysed regions and then returns regions in the given image that are likely to contain frontal face objects that the cascade has been trained for.

[5] http://www.intel.com/technology/computing/opencv/

Unlike faces, hands exhibit complex variable poses as seen from a camera and thus, a unique pose is difficult to unambiguously detect. In the proposed approach, this problem is solved by limiting the search space to a small volume. Once a face is detected, its 2D position from the input image is used as an index to find its 3D position using stereopsis. Based on this 3D information, an interaction volume is defined as shown in figure 18. Since the defined volume is a semi-spherical space in front of the face, objects that lie inside the volume are most likely to be human arms. Therefore by establishing the orientation of these objects, upper body pose has been determined.

If a hand/arm falls inside the 'action volume', a bounding box region of interest (ROI) is around it. An examination can then be performed to inspect 3D points within the ROI and thus establish the position of the arm(s).

3.9 Pointing Vector Acquisition

The algorithm used to derive the user's pointing direction is mainly based on the method proposed by Nicklel and Stiefelhagen [49] and Kelvin and Masahiro [125]. To determine the user's pointing direction, a line would be drawn from the user's eye through the user's pointing finger (eye-finger vector) that intersects with the screen. The resulting position on the display would be the user's intended target. An on-screen cursor would be shown at the same location at which the user is pointing (Figure 19).

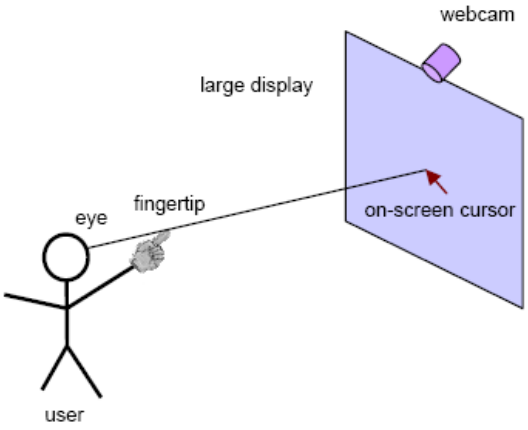


Figure 19: A webcam tracks the user's eye and fingertip. A line is extracted from these two points and extended to the display area where the on-screen cursor is calculated.

3.10 Optimal Camera orientation for Front Projected Large Displays

The method proposed in section 3.9 was initially tested with stereo cameras at eye level. However problems arise when a user points horizontally directly at one of the cameras (Figure 20) when interacting with the projected image. Furthermore, it is problematic having a camera occupying the centre of a screen in an interactive space. In order to solve this problem, a camera orientation study has been conducted to find a better camera orientation for a frontal public interactive display interaction.

The result is shown in chapter 5. It suggests that by mounting the camera above the screen, looking downward towards the user at an oblique angle is optimal for such screen interactions. In this way, a user pointing to the display is unlikely to point directly to the camera during interactions because the display is always at some distance below the camera. Nevertheless, this approach poses a new problem for the system because when an arm is extended out in front of the user, the disparity map would indicate that the head is closer to the camera (figure 21). Therefore 3D points that make up the scene must now be projected to a common coordinate frame in the body's frame of reference (rather than the real world frame of reference). This affine transformation can be achieved by using simple x rotation matrices (equation 13) to project each point back to the body frame of reference.

$$\Re_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (13)$$

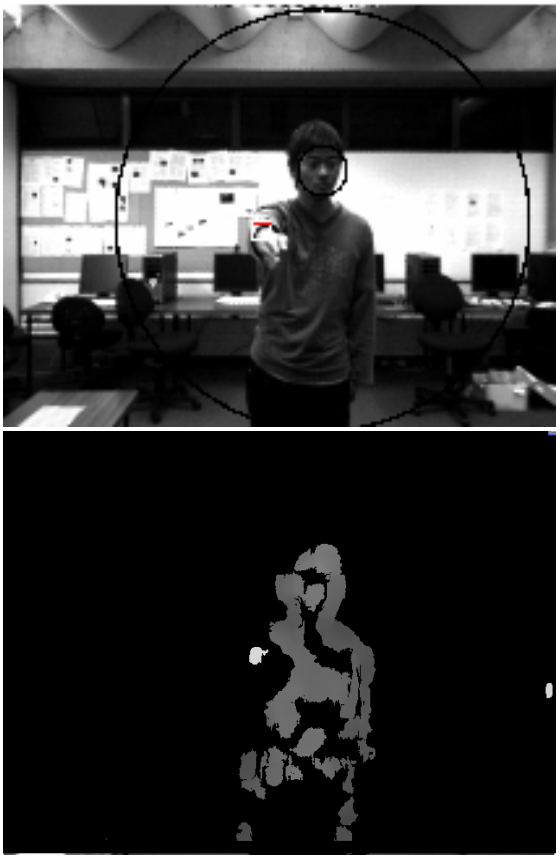


Figure 20. Result of a user pointing directly at the camera.

Problems arise when the user points directly at the camera. There was insufficient depth information for this pose.

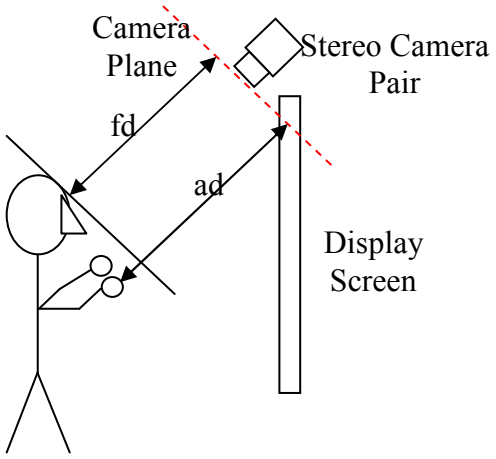


Figure 21. Optimal camera position. The distance between the head and camera (fd) is less than the distance between the arm and camera (ad).

3.11 2D Silhouette Algorithm for 3D Objects

Many human-computer interaction techniques exist using arm gestures, motions or a combination of both [3-11]. Different systems have also been proposed to recognize these gestures and motions[12-19]. However systems which rely on training or a 3D visual-hull reconstruction may seem complex and impractical. In this research, a new algorithm is proposed which does not rely on training or reconstruction of the 3D object. 3D data is simply interpreted in the 2D world.

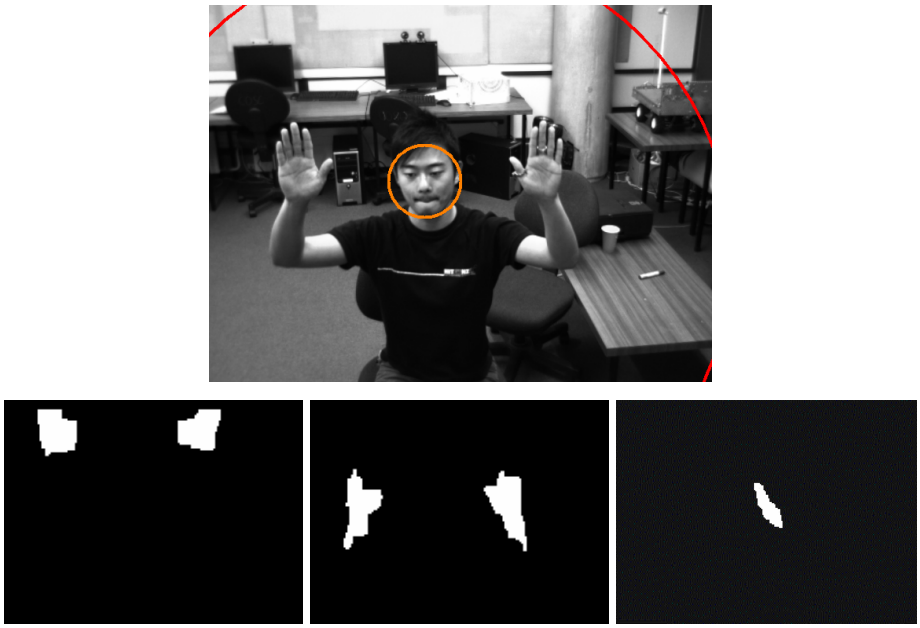


Figure 22. A user performing a pose and the corresponding three plane projections. Top: Original picture with face detected.
Bottom: the three plane projections after applying binary close.
Bottom left: XZ projection, Bottom middle: XY projection,
Bottom right: YZ projection.

First, 3D depth data of the upper limb is projected on to three planes: XZ, XY, and YZ planes. Figure 22 shows an example of a pose and its corresponding three plane projections. According to these three

projections, it is possible to understand the pose. XZ projection suggests that the pose was two handed with both arms held relatively close to the face. XY projection shows that the pose was vertical. The YZ projection suggests that both arms are parallel, because there is only one blob present.

When recognizing poses and tracking limb motions, each projection provides 2D information for the arm and is good for a set of gestures. Figure 22 bottom shows the three projection views and the 2D information they provide in 3D space. As shown in figure 23, when looking at the XZ projection, it is equivalent to looking at the pose from top-down. Hence, any horizontal poses or movements in the XZ projection are evidently shown. The XY projection is the same as looking at the pose frontally. Therefore, movements and poses parallel to the display screen would be shown clearly. The YZ projection is like viewing from the side. This projection is particularly useful when calculating if the position of one hand is higher than the other.

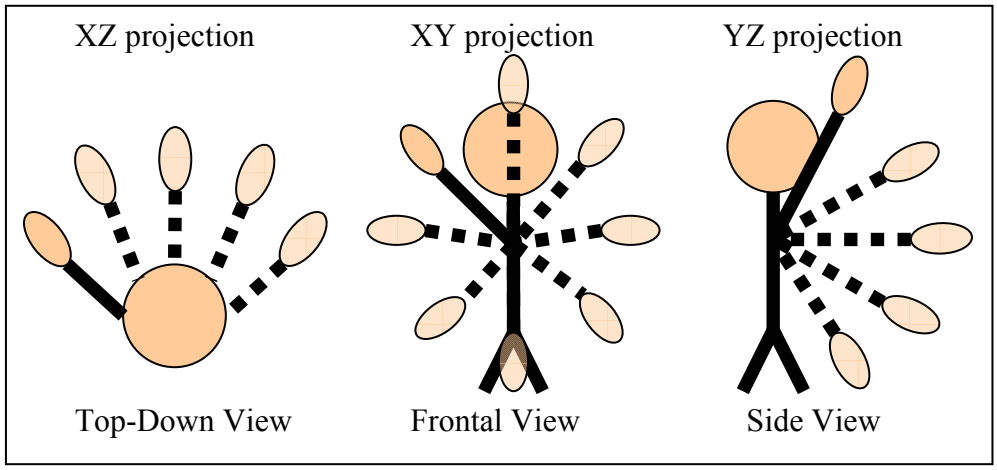


Figure 23. Three projection methods and their views

3.12 Noise Removal and Data Processing

While the images produced by 3D to 2D algorithm are comprehensible to humans, they are difficult for the system to analyse because there are too many fragmented parts, undesirable holes in the segments and scattered noise. Therefore, image processing is an essential step before it is fed to the connected component matching algorithm.

3.12.1 Median Filter

Median filter is the nonlinear filter used for remove the impulsive noise from an image [126-128]. This class of filter belongs to that of edge preserving smoothing filters which are non-linear filters. These filters smooth the data while keeping the small and sharp details.

Median filtering is a simple and very effective noise removal filtering process. Its performance is particularly good for removing shot noise. Shot noise consists of strong spike-like isolated values. It uses a 2-D mask that is applied to each pixel in the input image by centring it in a pixel, evaluating the covered pixel brightness and determining which brightness value is the median value. Figure 24 presents the concept of spatial filtering based on a 3x3 mask, where 'I' is the input image and O is the output image.

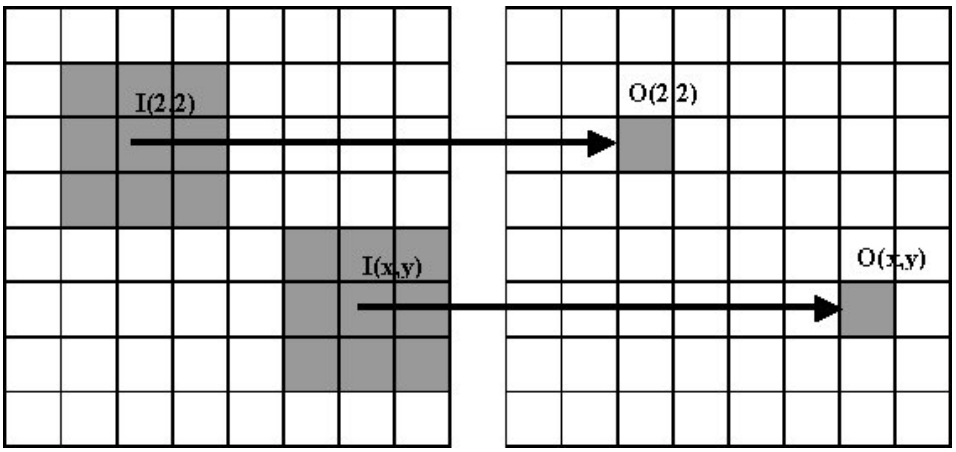


Figure 24. An example of a 3x3 median filter application

The median value is determined by placing the brightnesses in ascending order and selecting the centre value [126]. The obtained median value will be the value for that pixel in the output image. Figure 25 shows an example of the median filter application, as in this case, habitually a 3x3 median filter is used.

An example of median filtering of a single 3x3 window of values is shown below.

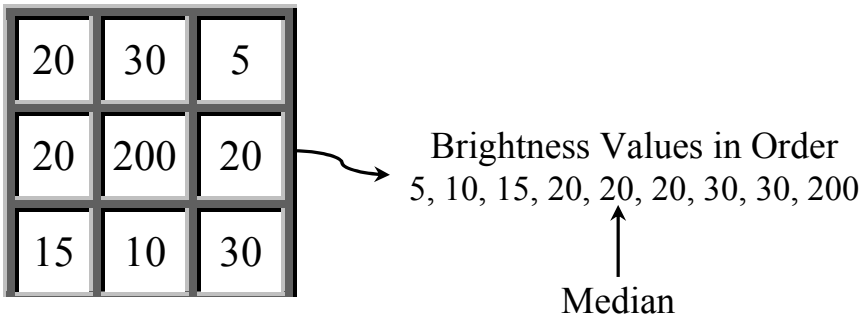


Figure 25. A 3x3 median filter mask. The median value is determined by placing the brightness in ascending order and selecting the centre value.

Consider the three binary projection images, their brightness value would consist of either 0 or 1. This simplifies the process of applying median filtering because the median value would be 1 if there are more 1s than 0s. For example, for a 3x3 median filter, if there are five 1s in the nine grids, the median value would be 1. There is no need to sort nine values into order.

3.12.2 Morphological Filters

There are four basic morphology filters: erosion, dilation, opening and closing. The equations are as follows.

$$E(A,B) = A \otimes B \tag{14}$$

$$D(A,B) = A \oplus B \tag{15}$$

$$O(A,B) = A \circ B = D(E(A,B),B) = (A \otimes B) \oplus B \tag{16}$$

$$C(A,B) = A \bullet B = E(D(A,B),B) = (A \oplus B) \otimes B \tag{17}$$

In the case of filling gulfs, channels and lakes in an image, a binary closing filter is normally applied.

3.13 Connected Component Matching

In this research, the information explained in 3.11 is exploited by classifying individual arms by connectivity. This is achieved by applying connected component labelling on each of the three projection binary images.

A description of the algorithm is as follows [129]. At each stage, the algorithm is scanning through two consecutive rows, termed *LastRow* and *ThisRow*. During this scan, it sees a region in each row. In Cases 1-4, the region in *LastRow* starts two or more columns before the region in

ThisRow. In Case 1, the region in *LastRow* ends one or more columns before the region in *ThisRow* starts. Therefore, these regions are not connected. In Case 2, the region in *LastRow* starts before the region in *ThisRow*, but the *LastRow* region continues at least to the column just before the region in *ThisRow* starts. Or it continues further, but it ends before the region in *ThisRow* ends. Therefore, if these regions have the same colour, then they are connected.

Based on this algorithm, the OpenCV blob extraction library provides the main functions such as extracting 8-connected components in binary images and then filtering the obtained regions to get the interest objects in the image. Figure 26 top shows the recognized blobs from XY projection in figure 22 bottom middle.

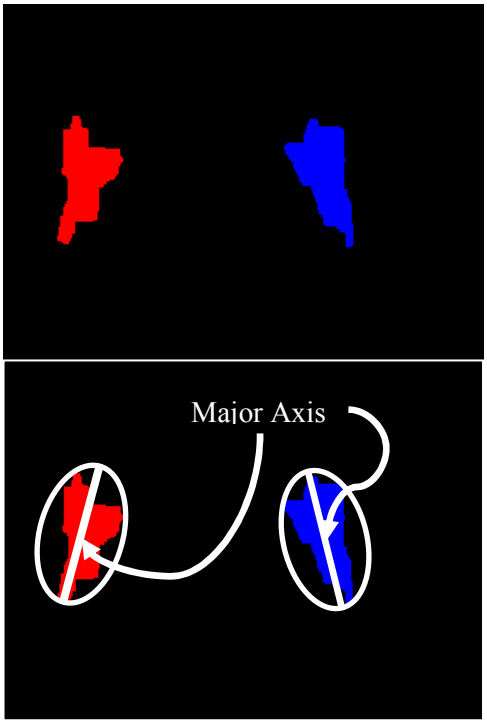


Figure 26. Result of applying connected component matching algorithm to the projection image and the major axis. Top: Connected regions in the XY projection are recognized as the red and blue blob. Bottom: The

resultant bounding eclipse for each of the blobs and their corresponding major axis.

After labelling each blob from the projection images, a bounding eclipse is calculated for each of the identified blobs. For every eclipse, there exists a major axis, which is the longest axis which can fit within the eclipse; and an eclipse centre, which is useful when tracking the location of the arms.

The orientation of the arms can be derived by calculating the angle between the major axis and the horizontal x axis. Nevertheless, calculating the angle between these two lines could result in two angles. For example, two straight lines with a 45 degree angle is equivalent to two lines with a 315 degree angle. Therefore there is a need to distinguish between the two angles.

The projection image is divided into four divisions as shown in figure 27. The horizontal and vertical dividers cross just below the face at the height of the shoulder. The correct angle for the arm is determined according to the location of the blob. If the centre of a blob lies in division one, as shown in figure 27, it is expected to have an angle within the range of 0 to 89 degrees. Similarly, 90 to 179 degrees for division 2, 180 to 269 degrees for division 3, and 270 to 359 for division 4.

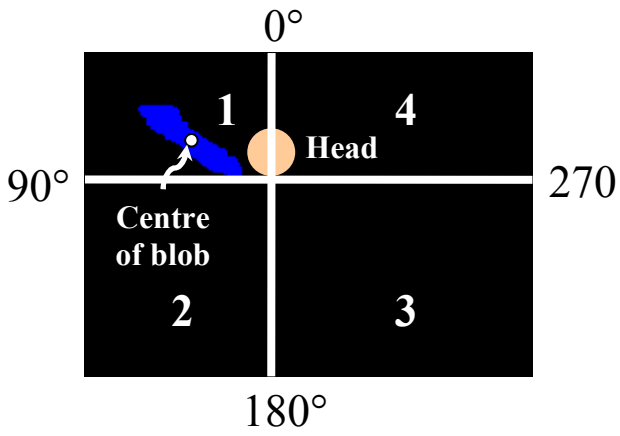


Figure 27. The four divisions of a projection image.

3.14 Closed-World Tracking

3.14.1 Introduction

A connected component matching algorithm is convenient for recognizing blobs from an image. However, the order of blob recognition is not guaranteed to be the same. For instance, the blob recognized for the left arm may be recognized as a blob for the right arm in the next frames. Therefore there is a need to track these identified blobs. In this research, closed- world tracking [130] is used to solve this problem.

Strat [131] has demonstrated that context-dependent visual routines are powerful tools for image understanding in complex static domains. Context is one way of addressing the knowledge-selection problem in dynamic, multi-object tracking. Consider the context of a tracking problem to be a boundary in the space of knowledge — a boundary outside of which knowledge is not helpful in solving the tracking problem[130].

In the domain of the three projection images, a context could be:

"An empty region that contains only two hands; one hand has 300 pixels at position A and was last moving left; the other hand has 200 pixels at position B and was last moving right."

3.14.2 Closed-World Assumption

One way to exploit such contextual knowledge is to use a closed-world assumption. A closed-world is a region of space and time in which the specific context of what is in the region is assumed to be known. It limits the knowledge relevant to tracking at a given instant and therefore reduces the complexity of the tracking problem.

In this research, the tracking algorithm has four data structures: (1) Each arm in the world has a data structure that stores the arm's size, velocity, current and past positions. This information is used for matching each object in the last frame to a blob in the new frame. (2) A local closed-world data structure exists for every blob, as in [130] and stores which objects are assigned to the blob and how long they have been there. Two objects that are touching will appear as one blob; therefore, they should be assigned to the same closed-world. The state of the closed-world to which an object is assigned determines the way the object's properties are re-estimated from the current frame. (4) Finally, the system uses knowledge about the global closed-world, which stores information about which objects are in the entire scene.

At each time step, the algorithm must match objects in the last frame to blobs in the current frame using the object and blob properties such as size and position. Once all objects are matched to blobs, the object

properties are updated using the new blob information and the state of the new closed world.

In the tracking process, three properties are computed for each object — position, velocity, and size — and used to compute matching distance measures. The first measure is the Euclidean distance (equation 18, distance d between two points p and q.) between an object's position and a blob position in the new frame. At high frame rates, objects are normally close to their blob in the new frame. The second measure is distance from predicted position. Velocity is estimated using an object's current and previous positions. Then the position of the object is predicted in the new frame and the Euclidean distance computed between the predicted position and the blob position in the new frame. Finally, the third measure is the size difference between an object's current blob and all blobs in the new frame, which should vary slowly at high frame rates.

$$d = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \qquad (18)$$

3.14.3 Enforcing hard constraints

Hard constraints can be used to prohibit a match between object 'i' and blob 'j' by flagging Mij as invalid. Three constraints currently used by the system are a maximum distance constraint and a size explanation constraint. The closed-world context controls which constraints are applied at a given stage in the matching process. The maximum distance constraint eliminates the possibility of any match between an object and a new blob that is greater than some reasonable maximum distance. In the proposed system of this thesis, the maximum distance is set at shoulder length from the shoulder point. According to anthropometric measures in section 4.4, a person's arm should not be further than 84cm away from the shoulder. For example, if a blob is more than 84 cm from the left shoulder of a person, the system should prohibit the labelling of 'left hand' to this blob. A second hard constraint can prohibit object-to-blob matches that are inconsistent with size information known about the objects already assigned to the blob. The third constraint used in the tracking process is the assumption that no two arm blobs shall touch. If there were two blobs in the previous frame, and they are now merged in the current frame, the tracking system is likely to be error-prone, therefore the tracking parameters are based on the previous frames. This is to simplify the problem domain and for accurate tracking performance.

3.15 Hand Grip Detection

Recognizing hand grips from the depth data has many advantages. It is one of the most intuitive ways of selecting objects. The implementation is simple, and it can be used as the key gesture to trigger other events. This is further explained later in this section.

A hand grip is modelled using three major components: the area of the blob from the projection image; the centre of the bounding eclipse; and the length of the minor axis, which is the shortest axis that is fitted within the eclipse and is perpendicular to the major axis. The area of the blob indicates the total area of the arm from the projection image; therefore a gripping action would trigger a decrease in blob area in the following frame. The minor axis indicates the breadth of the arm. Typically the minor axis length of a no-grip pose would be longer than that of a grip pose. The centre of the bounding eclipse is used to check if the arm is moving or not. This is important because owing to the noise of the stereo depth data, moving the arm from one place to another would usually change the size of the blob and the length of the minor axis. Even with a single mouse click, it is unusual to select an object while moving, so it is reasonable to recognize hand grip only when the arm is stationary.

A hand grip is recognized if the area of the blob and the length of the minor axis decreased dramatically from the previous frame, yet the centre of the bounding eclipse remained at the same position. Similarly a release of the grip is detected if the blob area and minor axis length increase while the eclipse centre stays in the same location. Figure 28 shows three consecutive frames from both a grip and a grip release.

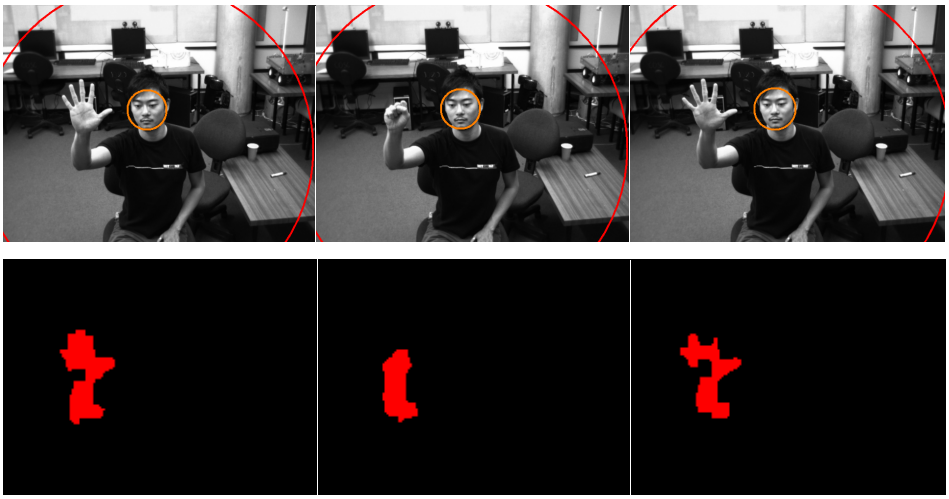


Figure 28. Three consecutive frames of a grip and a grip release and their corresponding XY projection images. Top row: 3 consecutive frames of a grip and a grip release. Bottom row: the corresponding XY projections of the three frames.

3.16 Acquisition of Principal Axes

The proposed system treats human arms as rigid objects. The position of a rigid body can be described by a combination of a translation and a rotation. Each can be represented by a vector. The angular position is also called orientation. There are several methods to describe numerically the orientation of a rigid body. In general, if the rigid body moves, both its linear and angular positions vary with time.

Principal axis is often used to describe a rigid body using angular displacements. (Figure 29). Therefore, acquiring principal axes of the human body provides useful information in understanding human poses. In the proposed system, the angular displacements are acquired from two projection images: XZ and YZ projections. By analysing the orientation of the blobs from the projected images as described in section 3.13, horizontal (φ) and vertical (θ) angular displacements are obtained.

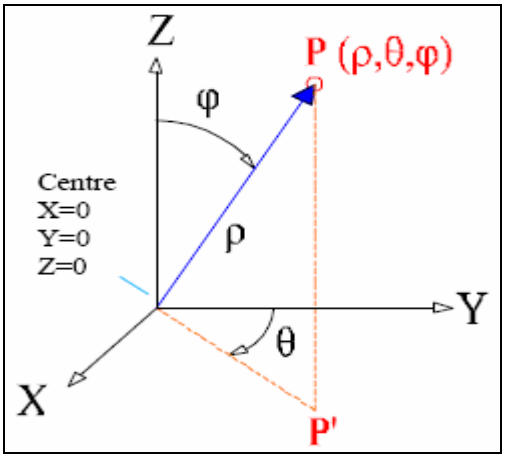


Figure 29. Two angle displacements for principal axis.

3.17 Summary

This chapter describes the proposed upper-limb detection algorithm implemented in this research. The algorithm is based on the stereo depth given by a single stereo camera. A cascade of Haar classifiers is used for human face detection and to account for activating upper limb tracking and deriving pointing vectors. Anthropometric constraints are taken into account together with the Zone of Convenient Reach (ZCR) to create an 'interaction volume' for every user interaction. It serves the purpose of upper limb segmentation as well as efficient user pose validation. One advantage of this proposed algorithm is the use of three orthogonal projection images obtained from projecting 3D point cloud of user arms onto XY, XZ and YZ planes. The use of this technique makes complicated 3D information interpretable in a straight forward 2D world. Observing the arm blobs on these three images produces sufficient information to recognize and track user upper limb movement. The principal axis of the arm and user poses such as hand grips can also be derived from the projection images. Morphological filters and median filters are explained in detail as they are essential for optimising image processing in this research. A stereo camera location and orientation study have been carried out and the result is shown in chapter 5.

4. Proposed Complete Optimal Set of Interactions for a Large Public Display

4.1 Introduction

As inspired by the direct manipulative gestures used in Apple's Ipod touch and Iphone, this chapter proposes a complete optimal set of interactions which can be applied to very large screen interactions. This approach is advantageous as the direct manipulation techniques are mostly in common use and therefore are intuitive to use. Section 4.2 describes in detail, the implementation for emulating such techniques using an arms pose and movements for large screen interactions. Section 4.3 outlines an experimental emulation of the traditional semaphore signalling system. The goal for such emulation is to evaluate the accuracy of arm orientations. The result of this experiment is shown in chapter 6.7.

4.2 Direct Manipulative Gestures

Multi-touch has received a great deal of attention recently through the widely disseminated research of Wilson [132, 133] and Han [134], and products such as the Apple iPhone[6] and now the Microsoft Surface [7][135]. Multi-touch has an even longer history however, and the first systems appeared well over two decades ago (see [136] for an overview of the major landmarks).

[6] Apple iPhone Multi-touch, http://www.apple.com/iphone
[7] Microsoft Surface, http://www.surface.com

One technique for detecting multiple fingertips on a display is to build custom sensing electronics into the surface itself [92, 137-139]. These systems are typically based on capacitive sensing, although other sensors can be utilized [138, 140]. They usually sense at low resolutions and are visually opaque, relying on projection for display. Even with this low-resolution sensing, rich sets of interactions have been demonstrated [92, 141]. What is harder with such systems (as they are non-optical) is to image the entire hand or other arbitrary physical objects close to or touching the surface. Rather, other objects apart from the hand need to be actively tagged to be detected by the surface [92, 142].

Camera-based systems allow more flexibility in sensing, providing a higher resolution optical system for capturing richer information about arbitrary objects in front of the display. Wilson [133] clearly highlights the tradeoffs of this flexibility, in terms of the high computational costs, the difficulty in achieving real-time interactive rates, ambiguity of data (particularly detecting when an object is hovering as opposed to touching the surface), and susceptibility to occlusion and adverse lighting conditions. This makes developing such systems an interesting and challenging problem.

Direct manipulation is a human-computer interaction style which involves continuous representation of objects of interest, and rapid, reversible, incremental actions and feedback. The intention is to allow a user to directly manipulate objects presented to them, using actions that correspond at least loosely to the physical world. Having real-world metaphors for objects and actions can make it easier for a user to learn and use an interface (some might say that the interface is more natural or intuitive), and rapid, incremental feedback allows a user to make fewer errors and complete tasks in less time, because they can see the results of

an action before completing the action. An example of direct-manipulation is resizing a graphical shape, such as a rectangle, by dragging its corners or edges with a mouse.

In early 2007, Iphone and Ipod touch were designed and marketed by Apple Inc[89]. They both have multi-touch screens with a virtual keyboard and buttons (Figure 30). The interface enables the user to move the content itself up or down by a touch-drag motion of the finger, much as one would freely slide or flick a playing card across a table with a finger. Similarly, scrolling through a long list in a menu works as if the list is pasted on the outer surface of a wheel: the wheel can be "spun" by sliding a finger over the display from bottom to top (or vice versa). In either case, the object continues to move based on the flicking motion of the finger, slowly decelerating as if affected by friction. In this way, the interface simulates the physics of 3D objects, giving it a real world feel.

Figure 30. Left: Iphone; Middle: Ipod Touch; Right: Virtual keyboard on the touch screen.

[8] http://en.wikipedia.org/wiki/Iphone
[9] http://en.wikipedia.org/wiki/Ipod_touch

The photo album and web page magnifications are examples of multi-touch sensing. It is possible to zoom in and out of web pages and photos by placing two fingers (e.g. thumb and forefinger) on the screen and spreading them further apart or closer together, as if stretching or squeezing the image. As can be intuitively expected from multi-touch sensing, the two fingers do not have to be from the same hand.

The interface is also used by MacBook Air, MacBook Pro and it is likely to replace many of the keyboard and mouse functions in the future. Multi-touch human-computer interface uses a multi-touch track pad; because it is a different mechanism to the normal computer mouse, it employs a different set of gestures.

Despite the popularity of a direct manipulation interface, no prior research has been carried out regarding the integration of such an interface with a large public interactive display system. The most common interfaces used in public interactive display systems have been based on background subtraction of objects in the display area. Although such interactive systems provide interesting and easy-to-learn environments for people to move virtual objects, however they are deficient in allowing complex interactions.

Table 5 shows a list of Ipod touch / Iphone gestures and their actions.

Ipod Touch / Iphone gestures	Action
Tap	To press or select a control or link (analogous to a single mouse click event).
Double Tap	To zoom in and centre a block of content or an image. To zoom out (if already zoomed in)
Flick	To scroll or pan quickly
Pinch open	To zoom in
Pinch close	To zoom out
Drag	To move the viewport or pan. (Analogous to a mouse drag event)
Slide	To unlock and confirm turning it off. The technique is also used for deleting files in certain screens such as videos, images and e-mails.
Two finger tap	Zoom out of a map quickly
Touch and hold	To display an information bubble, magnify content under the finger.
Two-finger scroll	To scroll up or down within a text area, an inline frame. (Analogous to a mouse wheel event).
Two-finger rotate	To rotate pictures clockwise and anti-clockwise in the photo album.

Table 5. Ipod Touch / Iphone gestures and their actions[10][11]

[10]http://ipodtouchnews.co.uk/2008/01/
[11]http://developer.apple.com/documentation/iPhone/Conceptual/iPhoneHIG/iPhoneUserEnvironment/chapter_2_section_5.html

This thesis advocates that the proposed system is adequate to be integrated with such direct manipulations and this would open the door to many more novel interactions in large public interactive systems. This notion is demonstrated in this research by emulating the existing direct manipulations present on Iphone and Ipod touch. This is further explained in section 5.2.2.

4.3 Emulation of Existing Direct Manipulative Gestures

In this research, ten out of twelve major interactions have been emulated. They are tap, double tap, two-hand tap, hold, drag, double drag, pinch open, pinch close, rotate clockwise, rotate anti-clockwise. The emulation is implemented based on three states. It is a non-deterministic automaton (NFA) and is described as follows.

Let M be the proposed DFA such that $M = (S, \sum, T, s, A)$, where

- S is a finite set of states.
  S= {q0, q1, q2}.
- Σ is a finite set of input the system takes, called the alphabet
  Σ = {1-hand grip, 1-hand release grip, 1-hand double grip, 2-hand double grip, 2-hand grip, 2-hand release grip, 1-hand drag, rotate clockwise, rotate anti-clockwise, pinch open, pinch close, 2-hand drag}.
- T is the transition function (S × Σ → S).
  T= {{q0 × '1-hand grip' → q1},
  {q0 × '1-hand double grip' → q0},
  {q0 × '2-hand grip' → q2},
  {q1 × 'drag' → q1},
  {q1 × '1-hand release grip' → q0},

{q2 × {rotate clockwise, rotate anti-clockwise, pinch open, pinch close, 2-hand drag} → q2},
{q2 × '2-hand release grip' → q0}}.

- 's' is the start state.
  s = q0.
- A is a set of accept states
  A = q0.

This is summarised in a finite state machine (FSM) graph (figure 31)

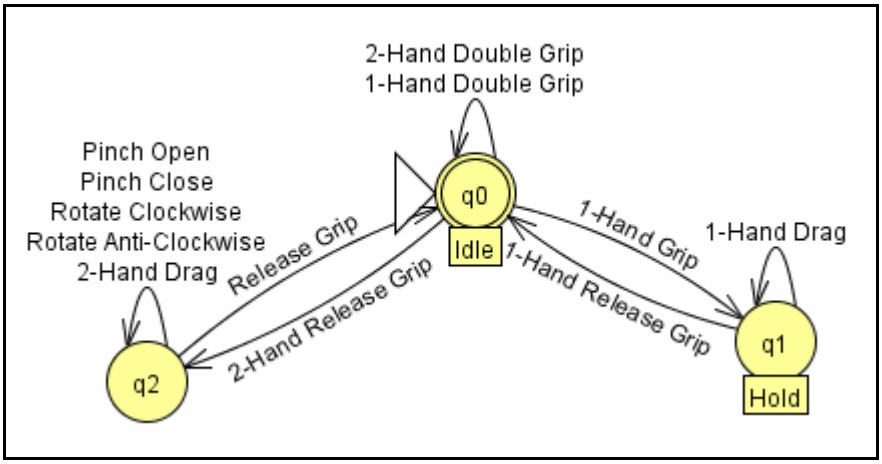


Figure 31. The FSM graph of the emulation.

Tap is modelled as a single grip, while double tap is implemented with two consecutive grip events. A drag event is recognized by a gripping pose, followed by a movement of the whole arm, and then the release of the grip. Pinch open and pinch close are similar, hence they are modelled together by two drag events from two arms at the same time. During dragging, if the distance between the centres of two arm blobs increases, it is classified as a pinch open; similarly, if the centres become closer, it is classified as a pinch close.

To disambiguate each input, motion gestures such as pinch open/close, rotate clockwise/anti-clockwise, 1-hand and 2-hand drags are required to be triggered using hand grip(s) and brought to closure by a release of the grip(s). This is modelled by accept state q0. The accept state helps the system to realise the error in the user input and/or tracking process and allows it to reset the state. For example, if a user performs a 1-hand drag but forgets to release the grip afterwards, or the release grip gesture was missed by the system, the current state should be reset to the start state q0 after some idle time interval, or when a new input other than 'drag' is received.

4.4 Semaphore Flag Signalling System

The semaphore flag signalling system [12] is a system for conveying information at a distance by means of visual signals with hand-held flags, rods, bare or gloved hands. Semaphores were adopted and widely used (with hand-held flags replacing the mechanical arms) in the maritime world in the early 1800s. Semaphore signals were used, for example, at the Battle of Trafalgar. This was the period in which the modern naval semaphore system which uses hand-held flags was invented. It is still used during underway replenishment at sea and for emergency communication, and also by the English coast guard service.

The flags are held, arms extended, in various positions representing each of the letters of the alphabet. The pattern resembles a clock face divided into eight positions: up, down, out, high, low, for each of the left and right hands (Figure 32). Six letters require a hand to be brought across the body so that both flags are on the same side.

[12] http://www.anbg.gov.au/flags/semaphore.html

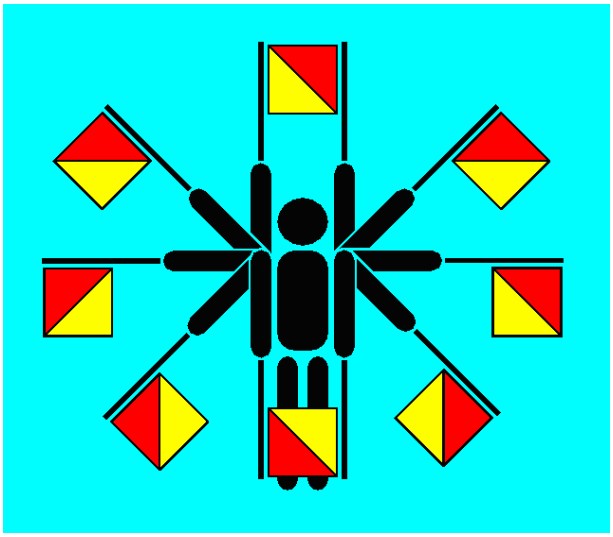


Figure 32. A clock face divided into 8 positions

Flag semaphore is an example of a sign language and is one of the old human to human interaction techniques. The gestures are simple and straightforward; therefore some gestures may be useful in a human to computer interaction. For example, error and cancel may be difficult to input via the traditional mouse, however, input is easy through simple arm gestures.

In this research, an application is implemented to recognize all 30 flag semaphores. It is mainly based on the observation of arms orientation from the frontal view. According to each angle of the blobs on the projection image; a corresponding flag semaphore signal is derived. Table 6 shows the list of flag semaphores and their implementation.

Flag positions	Meaning	Implementation
	Rest / Space	Arm 1 and arm 2 are both at 180°
	Numerals	Arm 1 is at 0° and arm 2 is at 315°
	Error	Arm 1 and arm 2 move up and down at the same time. Arm 1's angle increases with arm 2's angle decrease and vice versa. At least three iterations are necessary.
	Cancel	Arm 1 is at 45° and arm 2 is at 225°
	A / 1	Arm 1 is at 135° and arm 2 is at 180°
	B / 2	Arm 1 is at 90° and arm 2 is at 180°
	C / 3 Acknowledge	Arm 1 is at 45° and arm 2 is at 180°
	D / 4	Arm 1 is at 0° and arm 2 is at 180°

	E / 5	Arm 1 is at 180° and arm 2 is at 315°
	F / 6	Arm 1 is at 180° and arm 2 is at 270°
	G / 7	Arm 1 is at 180° and arm 2 is at 225
	H / 8	Arm 1 is at 90° and arm 2 is at 135°
	I / 9	Arm 1 is at 45° and arm 2 is at 135°
	J Letters	Arm 1 is at 0° and arm 2 is at 270°
	K / 0	Arm 1 is at 135° and arm 2 is at 0°
	L	Arm 1 is at 135° and arm 2 is at 315°
	M	Arm 1 is at 135° and arm 2 is at 270°
	N Negative	Arm 1 is at 135° and arm 2 is at 225°

	O	Arm 1 is at 90° and arm 2 is at 45°
	P	Arm 1 is at 90° and arm 2 is at 0°
	Q	Arm 1 is at 90° and arm 2 is at 315°
	R	Arm 1 is at 90° and arm 2 is at 270°
	S	Arm 1 is at 90° and arm 2 is at 225°
	T	Arm 1 is at 45° and arm 2 is at 0°
	U	Arm 1 is at 45° and arm 2 is at 315°
	V	Arm 1 is at 0° and arm 2 is at 225°
	W	Arm 1 is at 315° and arm 2 is at 270°
	X	Arm 1 is at 315° and arm 2 is at 225°

	Y	Arm 1 is at 45° and arm 2 is at 270°
	Z	Arm 1 is at 270° and arm 2 is at 225°

Table 6. A complete list of flag semaphores, their meanings and implementation.

When people perform gestures as precise as making particular angles using their arms, it is impossible make an accurate angle. The proposed application tolerates 45 degrees of inaccuracy for each of the eight positions of the clock face. (Figure 33) The coloured regions represent the tolerance for each of the eight divisions: positions at 0, 45, 90, 135, 180, 225, 270, 315 degrees. For example, if an arm falls between 22.5 and 67.5 degrees on the clock face, 45 degrees is taken for this arm.

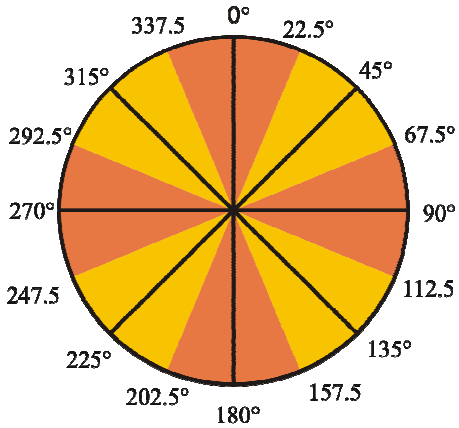


Figure 33. Coloured regions represent the tolerance for each of the eight divisions.

Among all 30 flag semaphores, one dynamic motion gesture, ‘error’ exists. This requires the user to raise and lower both of the arms together. If both left and right arms keep moving for three seconds and they move in such a way that when the angle of the right arm increases, the angle of the left arm decreases and vice versa; and if this cycle lasts for three iterations, an ‘error’ signal is recognized. The rest of the 29 semaphores are implemented based on the observation of static arm angles when they have remained stationary for two seconds.

4.5 Summary

This chapter proposed a complete optimal set of interactions for very large interactive displays based on the direct manipulation input as used in Apple’s Ipod touch and Iphones. Furthermore, such interaction technique provides intuitive gestures for user interaction. A detailed description of the emulation is described in section 4.3.2. The result of the evaluation is shown in chapter 5.5. This chapter also depicts the implementation of the traditional semaphore signalling system using the proposed algorithm. The purpose of such a signalling system is impractical to be applied to screen interactions, however is useful in terms of evaluation of the accuracy of arm orientation. The experiment evaluation is described in section 6.6.

5. Experimental Evaluations

5.1 Introduction

As mentioned in chapter 2, there is a large body of prior research into tracking body parts based on various skin hue manifold algorithms. In this research, a hue invariant approach is chosen using stereopsis. The proposed algorithm is evaluated in terms of accuracy and performance. A study has also been conducted to find the optimal camera location and orientation for a stereo camera based on a very large public interactive display environment.

The algorithms and application systems of upper-limb detection have been discussed in chapters 4 and 5. In this chapter, four experiments have been carried out to evaluate the proposed algorithm. These include experiments for pointing direction, principal axis angles, recognizing direct manipulation gestures and flag semaphores. Furthermore, the comfort in using the pointing system is ascertained by having participants subjectively rate the system using a questionnaire form which assesses aspects of operation, fatigue, comfort, and overall usability.

5.2 Hardware of the Experimental System

The experiment system was developed on a computer system with the following profile:

- CPU: 2 2.13GHz CPUs
- RAM: 1.98 GB Ram
- OS: Microsoft Windows XP Professional (Service Pack 2)
- Graphic Hardware: NVIDIA GeForce 8500GT

- 1 fire wire port

The parameters of the camera used are listed below:

- Point Grey Research's Digiclops Stereo system[13] model Bumblebee2 camera
- Baseline distance: 120.0 mm apart
- Image resolution: 640x480 pixels (24 bit RGB colour)

5.3 Camera Orientation Study

According to chapter 4, problems arise when a user points horizontally directly at one of the cameras when interacting with the projected image on the wall. Furthermore, it is clearly problematic having a camera occluding the image by locating it at the centre of a screen in an interactive space. To solve this problem, a camera orientation study has been conducted to derive an optimal camera location and orientation for public interactive display interaction.

The experiment consisted of two parts; the results were compared with respect to the robustness of the upper limb bounding box. Each part involved a person pointing to 13 different directions (longitude, latitude) with 20 attempts. The results have been recorded and categorized into 'bounded', 'partial bounding' and 'no detection'. The 'bounded' category includes both perfect bounding and over bounding of arm/hand; 'partial bounding' is where only part of the arm/hand is bounded; 'no detection' represents matches where no bounding boxes were drawn. These results are shown in Figure 34 and 35.

Figure 34 shows the result for the proposed method with stereo cameras at eye level. The approach worked well with most pointing postures.

[13] http://www.ptgrey.com/products/stereo.asp

However, as expected this approach worked poorly when the pointing poses were at horizontal (i.e. at positions (0, 75), (0, 45), (0, -45), (0, 0) and (45, 0)). The performance was worst when the user pointed directly to the camera (0, 0) where out of 20 attempts, the system only picked up 13 partial encapsulations (7 were not recognized). This was because when the user points to the camera directly, the arm and hand reveal the smallest area that can be seen by the camera and so there was not enough depth information available along the arm for calculating an unambiguous direction vector.

Figure 35 shows the result for the proposed method with stereo cameras mounted above and looking down at the user with a rotation of 25 degrees along the x-axis. In contrast to figure 34, results from this camera orientation showed a good performance even when the user was pointing horizontally. The worst performance happened at (45, 0) because this was close to pointing directly at the camera. The latter result can be remedied by locating the camera at a greater distance above the display.

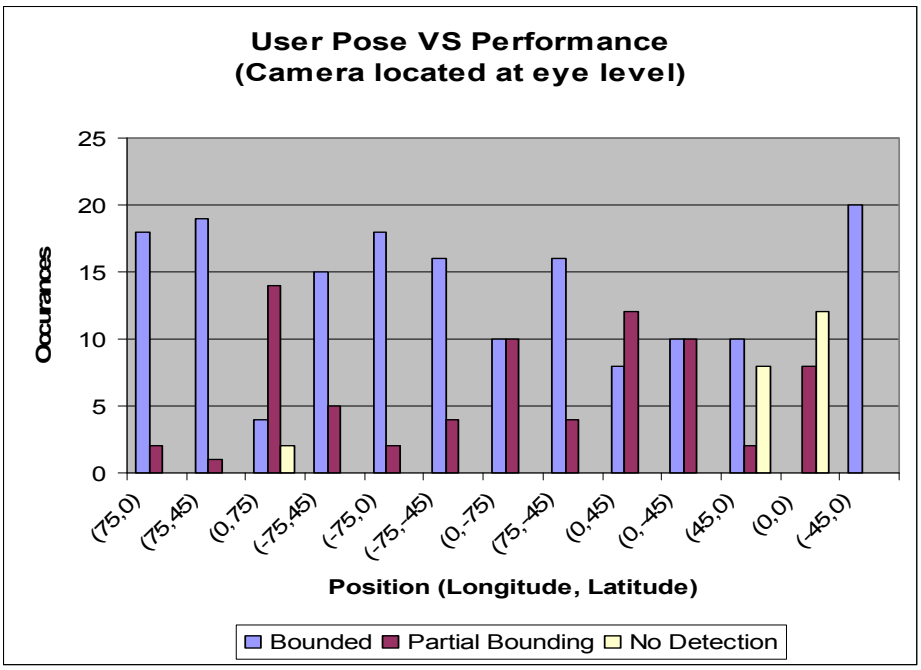


Figure 34. Performance for different pointing directions with camera located at eye level

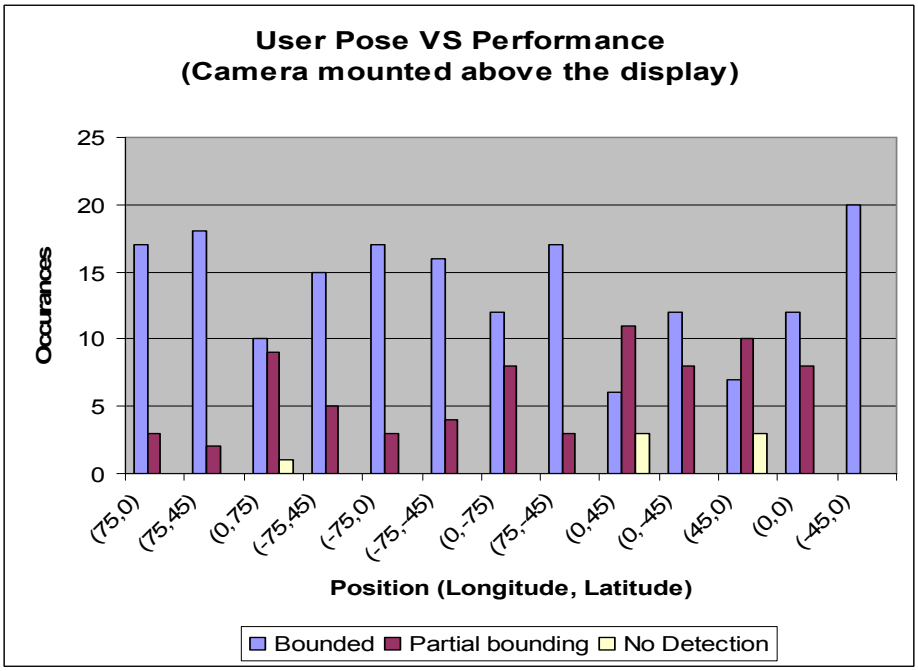


Figure 35. Performance for different pointing directions with the camera mounted above the display and rotated 25 degrees along the x-axis.

5.4 Pointing Experiment

In this section, the pointing performance of the proposed system was tested similarly to the experiment described in [143]. The experimental setup is shown in Figures 36 and 38.

5.4.1 Participants

Ten volunteer participants (six male, four female) were recruited from the local university campus. Participants ranged from 23 to 26 years (*mean* = 24). All were daily users of computers, reporting three to twelve hours usage per day (*mean* = 7). None had prior experience with arm pointing.

5.4.2 Apparatus

The experiment environment was based on the experiment system described in 6.2.1. The bumblebee stereo camera was mounted two metres above the ground. Input was via user's arm pointing directions.

5.4.3 Methodology

Participants were asked to stand three metres in front of the wall and point to 16 different points drawn on a 1x1m poster (figure 37). Each of the 16 points was drawn 20cm spaced both horizontally and vertically. The poster was attached to the wall one metre above the ground. A single stereo camera was mounted on the top of the poster looking down on the participants with an angle of 30 degrees. No visual feedback was given during these experiments; hence the user should have been unbiased and showing a natural pointing gesture.

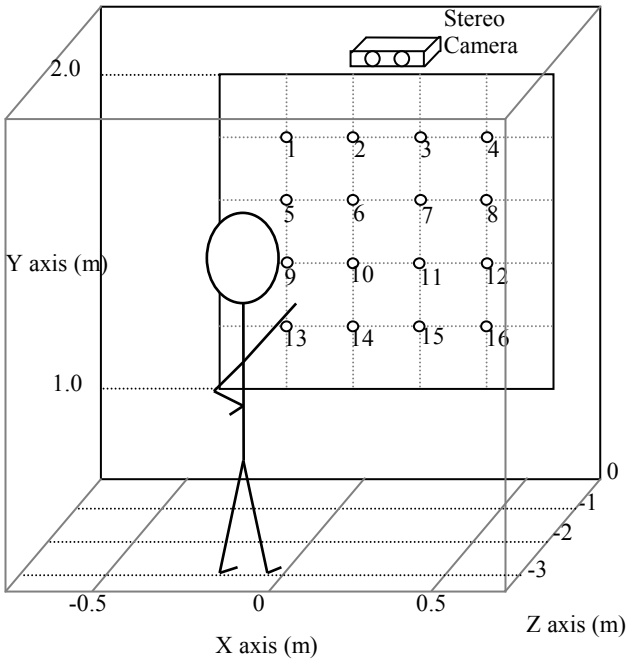


Figure 36. A sketch of the experimental setup.

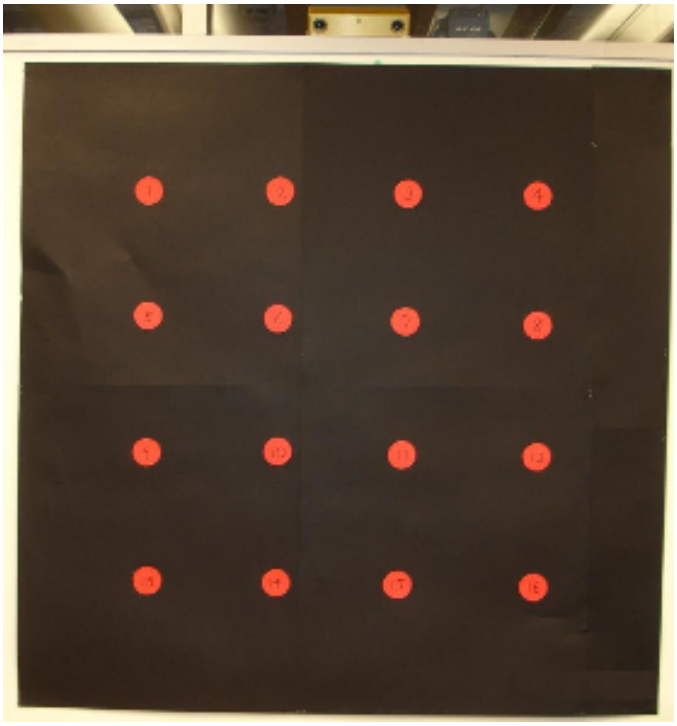


Figure 37. Poster and bumblebee stereo camera used in the pointing experiment.

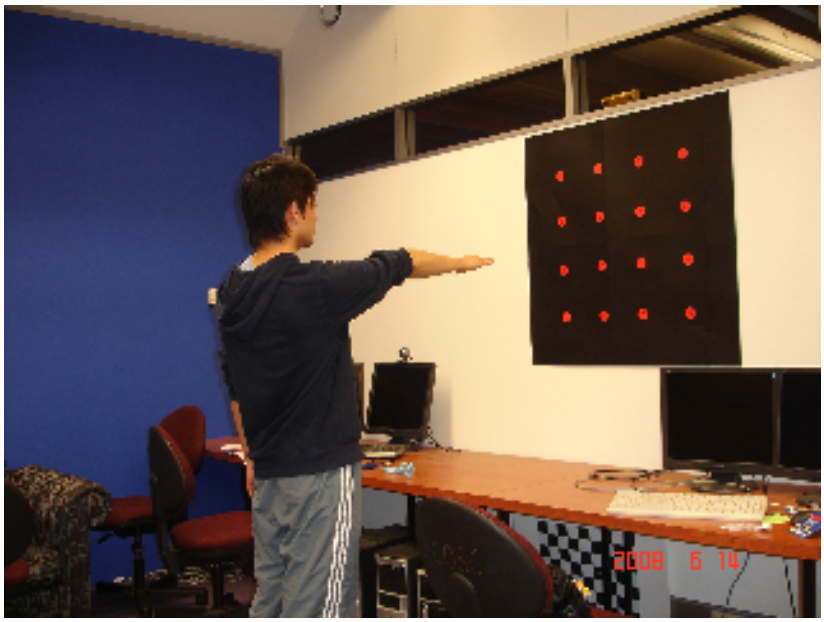
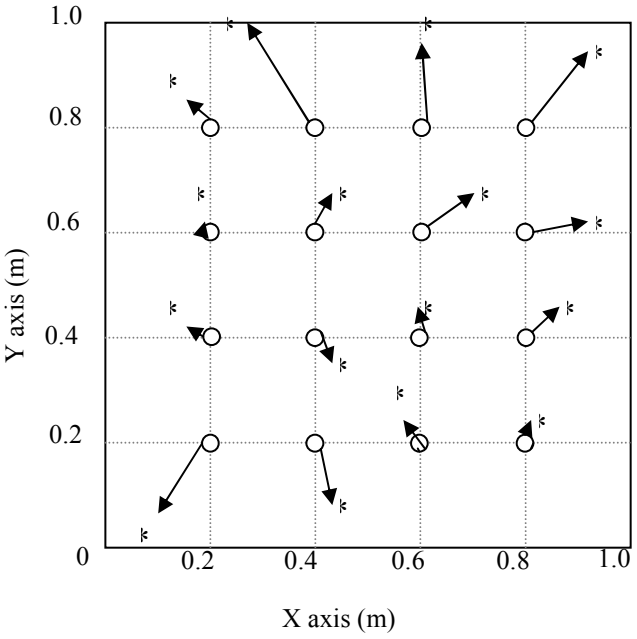


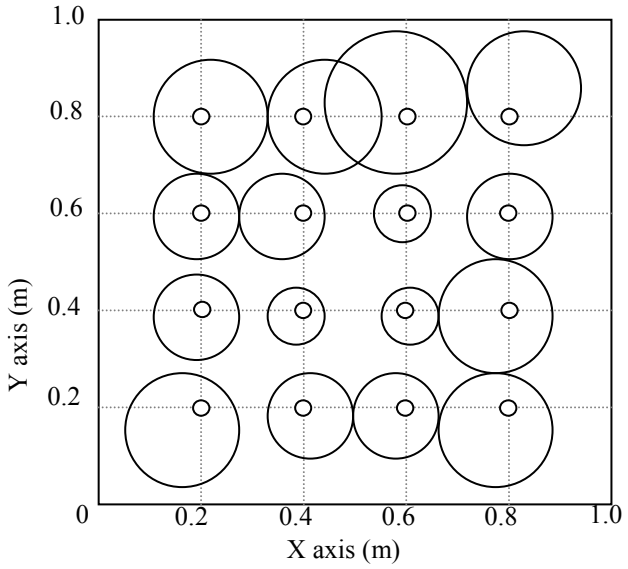
Figure 38. A person pointing at the red dots on the poster at a pointing experiment.

5.4.4 Result and Discussion

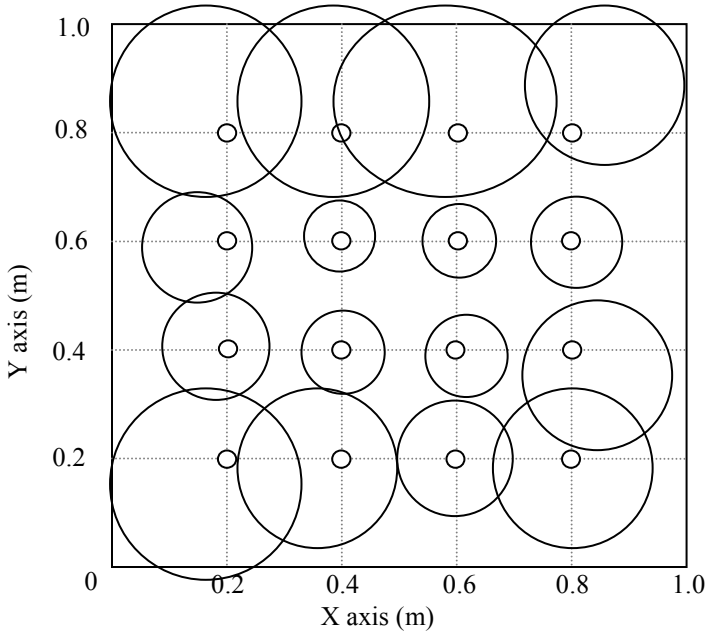
Figure 39 shows a representative pointing experiment. The circles (o) are the real positions displayed on the wall and the asterisks (*) connected by the arrows are the respective estimated positions at which the user is pointing. The positions 1 to 4 had the worst performance and the error was up to 0.2 m.


1.0
0.8
0.6
0.4
0.2
0
Y axis (m)
0.2
0.4
0.6
0.8
1.0
X axis (m)

a)


1.0
0.8
0.6
0.4
0.2
0
Y axis (m)
0.2
0.4
0.6
0.8
1.0
X axis (m)

(b)



(c)

Figure 39. Result from pointing experiments. a) Result of a representative pointing experiment. b) Average of errors. c) Maximum error.

Figure 39 b shows the average result of all ten experiments. Each circle illustrates the average error. In figure 39c, each circle indicates the maximum error. The total average error is 0.33 m and the maximum error is 0.43m.

According to the experiment result in [143], their system had the worst accuracy as the user points to the left such as positions 5 ,9, 10, and to lower positions 11, 12, 13, 14, 15, and 16. In comparison, the pointing system of this research is more accurate when the user points at those positions, however it is worse when the user points at positions close to the camera (positions 1-4).

The experiment result shows that the proposed system of this research is accurate at targets 5-12, which means the proposed system, is at its best performance if the target is between 1.4 – 1.6m above the ground.

The worst performance occurred at position 1-4; this is the result of the user pointing close to the camera's eye. In order to remove such a problem two further camera configurations can be evaluated:

1. Place two stereo cameras on either side of the top of the screen and:
2. Place three single cameras, one on either side of the top of the screen and one centred above the screen to constitute three disparity maps (from left-centre pair, right-centre pair and left-right cameras).

5.5 Direct Manipulative Gesture Experiments

This section describes the experiment for Iphone / Ipod touch emulation as described in Section 4.2. The purpose of this experiment is to show that the novel algorithm proposed by this thesis can be applied to modern screen interactions, including public interactive display systems.

5.5.1 Participants

Ten volunteer participants (six male, four female) were recruited from the local university campus. Participants ranged from 23 to 26 years (*mean* = 24). All were daily users of computers, reporting three to twelve hours usage per day (*mean* = 7). None had prior experience with arm pointing.

5.5.2 Apparatus

The experiment environment was based on the experiment system described in 6.2.1. The bumblebee stereo camera was mounted 1.8 metres above the ground. Output was presented on a 19" monitor 0.9 metres above the ground and directly under the camera. Input was via the user's arm orientation, size and width.

5.5.3 Methodology

The participants were asked to sit three metres in front of the camera and perform individual direction manipulation gestures / motions. When a user performed a gesture, the feedback was shown on a 19 inch display located underneath the camera. Figure 40 shows the experimental environment.

Figure 40. A person making a 'grip' gesture in a direct manipulation experiment.

The experiment consisted of ten blocks, each with five trials. Each of the ten blocks corresponded to ten Iphone / Ipod touch gestures described in section 5.2. These included: grip (1 hand), grip (2 hands), drag (1 hand), drag (2 hands), double click (1 hand), double click (2 hands) pinch open, pinch close, rotate clockwise, rotate anti-clockwise.

5.5.4 Result and Discussion

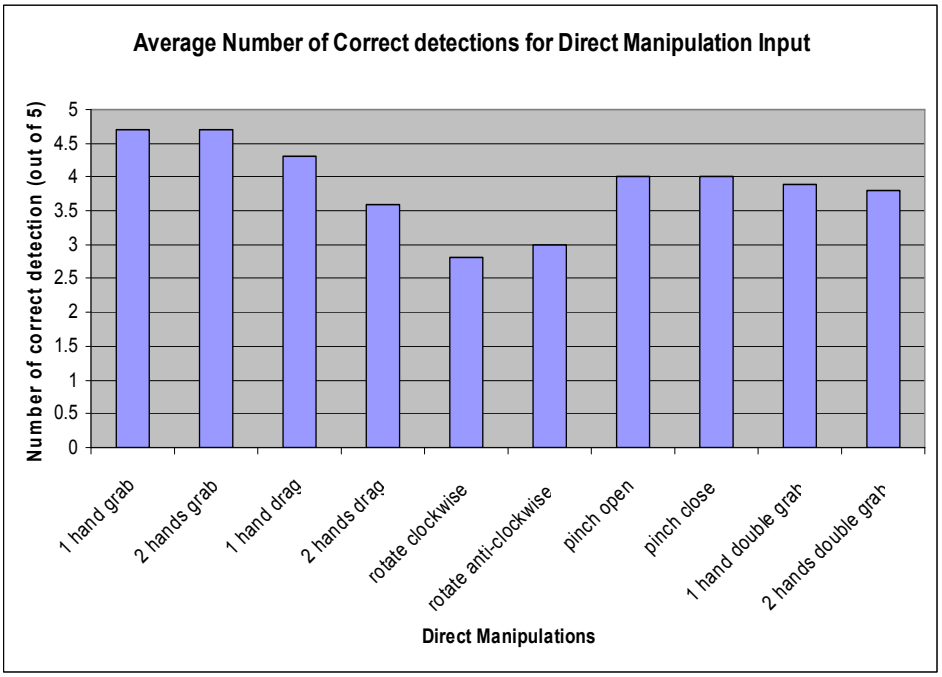


Figure 41. Average number of correct detections for the direct manipulation experiment.

Figure 41 shows the result of the experiment. Grab (1 hand) and grab (2 hands) had the highest number of correct detections. They both had 94% of accuracy. Input such as drag (1 hand), drag (2 hands), pinch open, pinch close, double grab (1 hand), and double grab (2 hands) had considerably good accuracy ranging from 72% to 86%. In contrast, rotate clockwise and rotate anti-clockwise had the worst accuracy of 56% and

60% respectively; this is due to the complexity of the input that both rotate clockwise and rotate anti-clockwise require the user to use both arms and perform good circular movements. Another reason for the bad performance is that both of these gestures needed the user to first perform a grab (2 hands) gesture first. Grab (2 hands) has the accuracy of 94 %, and 6% of the error rate contributes to the missed detection for both rotate clockwise and rotate anti-clockwise inputs.

The result shown in section 6.5.1 suggests that the accuracy for grab (1 hand) is as accurate as in past research (90% - 100%) [144-146]. Compared to the experiment in [147], their accuracy for recognizing zoom in (pinch open) and zoom out (pinch close) were 88% and 95 % accuracy. The system proposed by this research had a worse accuracy of 80 %. Unfortunately, there were no comparable results published for the remaining seven gestures.

The lowest accuracy occurred when rotate clockwise and anticlockwise were performed. These two are motion gestures and the performance may be improved by employing motion classifiers or a combination of classifiers [148].

5.6 Flag Semaphore Experiment

The purpose of the experiment described in this section was to test the accuracy of arm angle detection in the XY projection image. The experiment was based on a system described in section 5.3, which is designed to recognize all 30 flag semaphores according to the angle of the arms. This includes alphabets A-Z (numerals, 0-9, letters, negative), error, cancel, rest/space. A complete list of flag semaphore poses is presented in section 5.3.

5.6.1 Participants

Ten volunteer participants (six male, four female) were recruited from the local university campus. Participants ranged from 23 to 26 years (*mean* = 24). All were daily users of computers, reporting three to twelve hours usage per day (*mean* = 7). None had prior experience with arm pointing.

5.6.2 Apparatus

The experiment environment was based on the experiment system described in 6.2.1. The bumblebee stereo camera was mounted 1.8 metres above the ground. Output was presented on a 19" monitor 0.9 metres above the ground and directly under the camera. Input was via user's arm orientation.

5.6.3 Methodology

In this experiment, participants were asked sit three metres in front of the camera and perform all semaphore poses five times. When a user performed a gesture, the feedback was shown on a 19 inch display located underneath the camera. Figure 42 shows a person performing the letter 'U' at the experimental environment.

Figure 42. A person performing the letter 'U' gesture in the flag semaphore experiment.

5.6.4 Result and Discussion

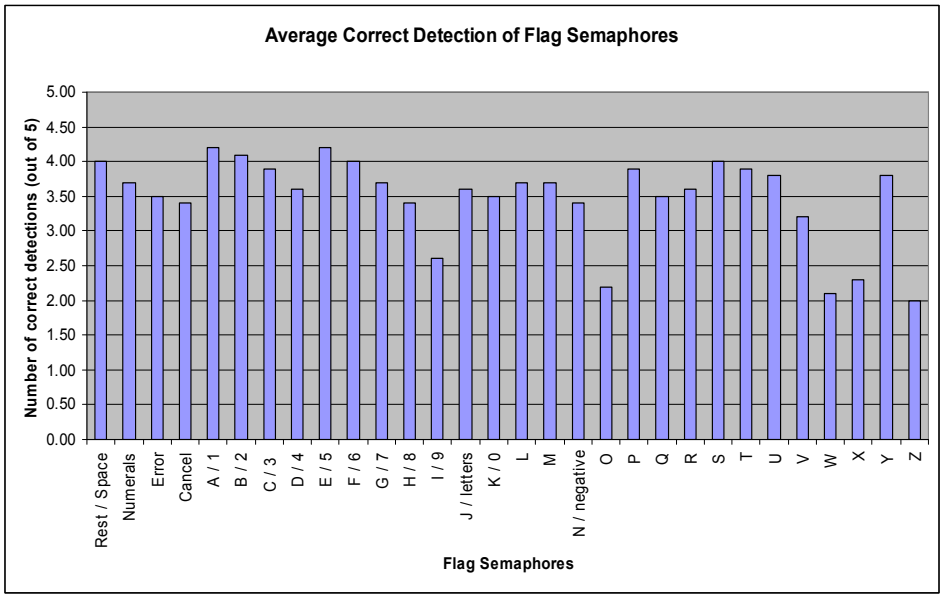


Figure 43. Average correct detection of flag semaphores.

The result shows that this application has an average correct recognition rate of 3.48 out of 5 attempts. This is 70% accuracy. According to figure 43, the semaphores that had the worst detection were ‘H/8’, ‘I/9’, ‘O’, ‘W’, ‘X’ and ‘Z’. These are the only six letters out of the whole set of semaphores which required the hand to be brought across the body so that both hands are on the same side. For users, it is awkward to perform these six letters while their bodies remain facing forward. After discarding the result of these six gestures, the overall accuracy for recognizing arm angles based on XY projection images was 75%.

This result suggests that for screen interactions, especially large displays, gestures which require the user to bring their arm across the body or extend it in a non intuitive direction is inadequate and should not be employed.

5.7 Principal Axes Experiments

Section 6.6 describes the experiment for testing the accuracy of recognizing arm angles in the XY projection images. This chapter describes two experiments to evaluate the accuracy for determining arm angles in the remaining two images: XZ and YZ projection images. The angles on these two images contribute to the accuracy of the principal axis of the arm.

5.7.1 Participants

Ten volunteer participants (six male, four female) were recruited from the local university campus. Participants ranged from 23 to 26 years (*mean* = 24). All were daily users of computers, reporting three to twelve hours usage per day (*mean* = 7). None had prior experience with arm pointing.

5.7.2 Apparatus

The experiment environment was based on the experiment system described in 6.2.1. The bumblebee stereo camera was mounted 1.8 metres above the ground. Output was presented on a 19" monitor 0.9 metres above the ground and directly under the camera. Input was via the user's arm orientation.

5.7.3 Methodology

The two angles used to represent the principal axis of the arms were based on the observation of the arm orientation on both XZ and YZ projection images. Therefore the study consisted of two experiments. The first experiment required the user to stand on a fixed position and point to five horizontal directions angled at 30, 60, 90, 120 and 150 degrees along the x-axis. Similarly, the second experiment of the experiment required the user to point to five vertical directions angled at 60, 75, 90, 105, and 120 degrees along the y-axis.

The angles 30 and 45 degrees in the vertical experiment were excluded owing to the problem described in section 4.6. Pointing at these directions is too close to pointing at the camera; therefore the detected angles at these directions are usually undefined.

Pointing at specific angles is however difficult for human beings, since they can only point at directions with approximate angles. To help minimize the approximation errors caused by each participant, coloured tape stripes and green tags were used on both the ground and on the wall prior to the experiment to guide pointing during the experiment. Figure 44 shows the experimental environment.

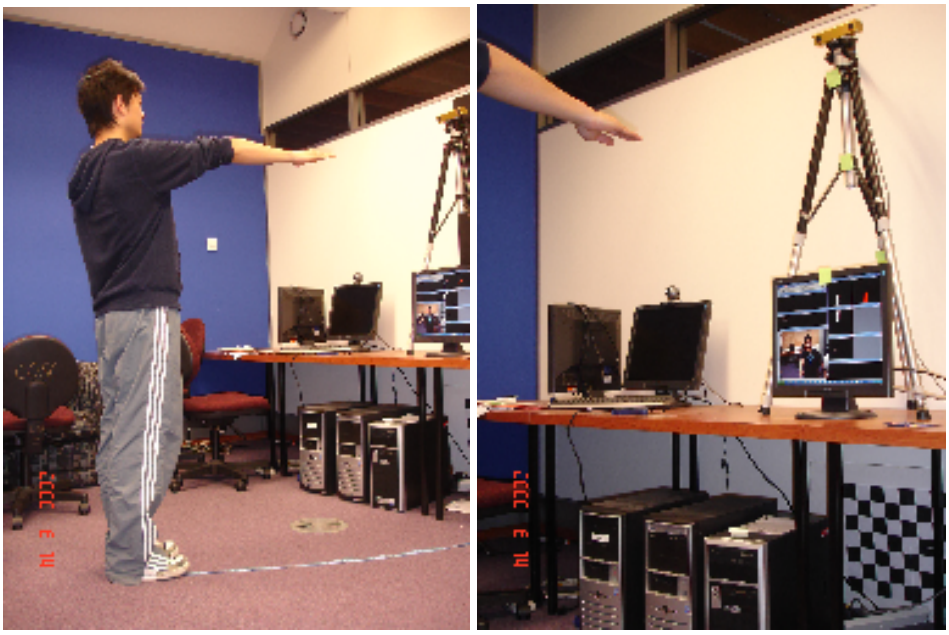


Figure 44. Experimental setup for principal axes experiment. Left: blue stripe (on the ground) used in the horizontal experiment. Right: green tags used in the vertical experiment.

5.7.4 Result and Discussion

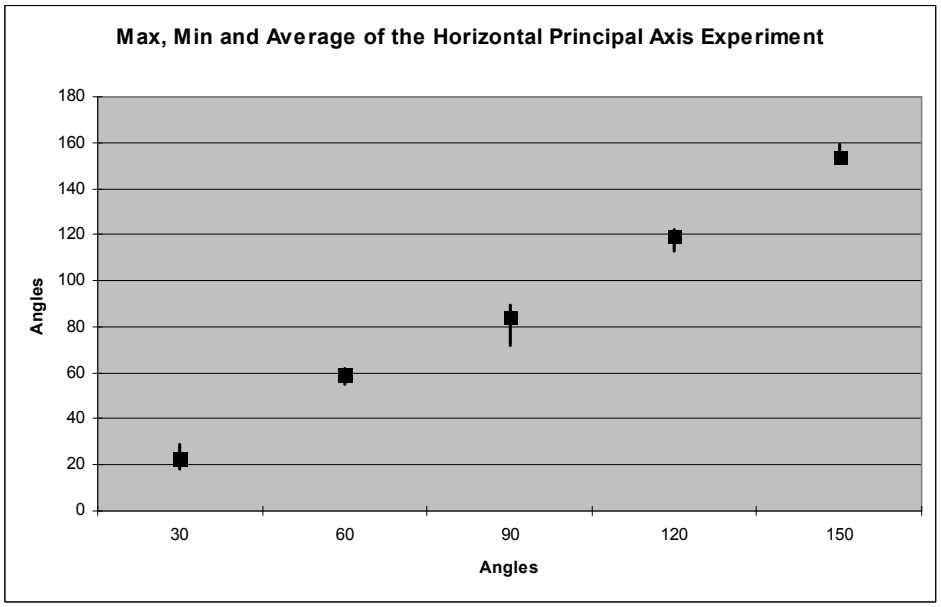


Figure 45. Result for the horizontal angle experiment.

The average detected angle for 30 degrees was 23.1 degrees, with maxima and minima of 29 and 18 degrees. The 60 degree angle has the average detected angle of 58.8 degrees, with maxima and minima of 62 and 55 degrees. The 90 degree angle has the average detected angle of 84.1 degrees with maxima and minima of 89 and 72 degrees. The 120 degree angle has the average of 119.8 degrees with maxima and minima of 122 and 113 degrees. The 150 degree angle has the average detected angle of 154 with maxima and minima of 159 and 151 degrees.

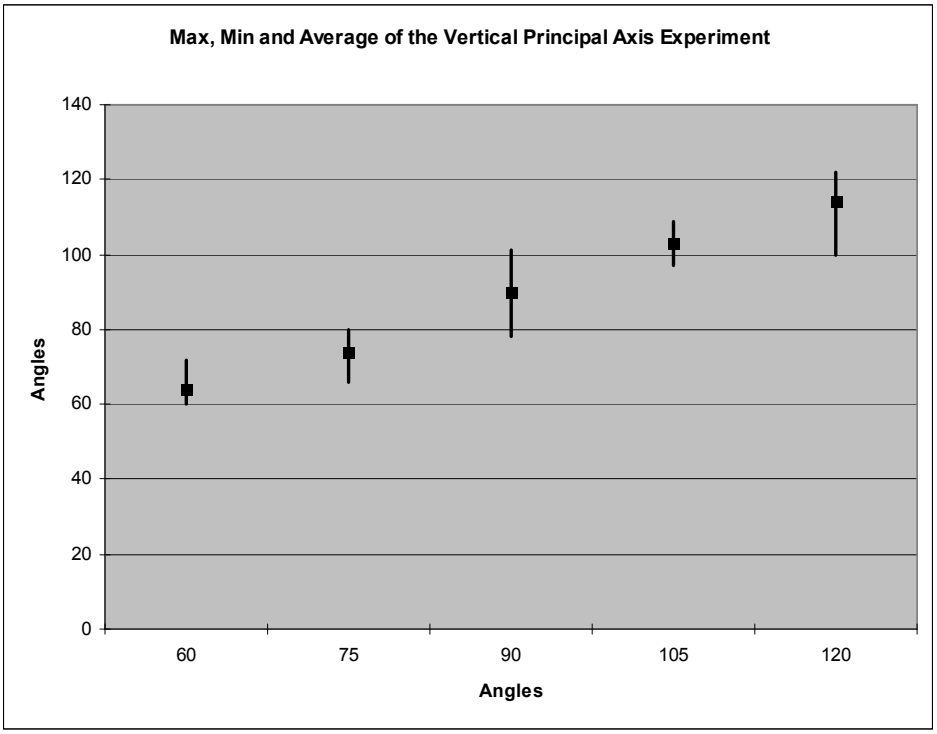


Figure 46. Result for the vertical angle experiment.

The average detected angle for 60 degrees was 64.3 degrees, with maxima and minima of 72 and 60 degrees. The 75 degree angle has the average detected angle of 73.9 degrees, with maxima and minima of 80 and 66 degrees. The 90 degree angle has the average detected angle of

89.7 degrees with maxima and minima of 101 and 78 degrees. The 105 degree angle has the average of 103.1 degrees with maxima and minima of 109 and 97 degrees. The 120 degree angle has the average detected angle of 114.1 with maxima and minima of 100 and 122 degrees.

According to figure 45, the best performance occurred when the user pointed horizontally at 60 and 120 degree directions. The worst performance occurred at the 90 degrees direction. It has the highest marginal error of 17 degrees. Figure 46 shows the best performance occurred when the user pointed vertically at 60, 75, and 105 degrees. They also have the fewest marginal errors. The worst performance occurred at directions 90 and 120 degrees.

There is a similar experiment measuring the accuracy of angle orientation of the arm described in [149]. As opposed to the method proposed by this research, of determining the arm angles by interpreting the 3D data on 2D projection planes, the system in [149] judges the angles based directly on the 3D depth data. In their experiment, the average for the mean angle displacements of the arm angle in XZ plane is 5.92 and 7.38 degrees for the XY plane. In comparison, the result for the proposed system shows that the average for the angle displacements in XZ and YZ planes are 3.64 and 2.9 degrees. The result suggests that interpreting 3D data in 2D form is indeed more accurate than interpreting directly on noisy 3D depth data.

5.8 Conclusion

This chapter describes a camera orientation study, a direct manipulation emulation experiment, a flag semaphore signalling system experiment, and two principal axis experiments.

The camera orientation study result shows that by mounting the camera position some distance above the projected display and looking down at an oblique angle at the user would successfully reduce the number of 'partial bounding' and 'no detection' rates as opposed to the camera located at eye level (centre of the display screen).

The result of the pointing experiment suggests that the performance of the proposed pointing system is accurate and improves on prior research [143] at positions 5, 9, 10, 11, 12, 13, 14, 15 and 16. However the performance of the proposed system was worse when the user's pointing direction was close to pointing directly at the camera, i.e. positions 1-4.

Most of the inputs in the direct manipulation experiment had considerably good accuracy (72% to 94%). The result shows that it is feasible to employ such interaction methods for a large interactive display while still providing reasonably accurate detection.

The flag semaphore experiment was designed to evaluate the accuracy of the arm orientation from the XY projection image. The result shows that the average accuracy was 70 % and the worst detection occurred when participants performed gestures which required the hand to be brought across the body so that both hands were on the same side. The result also suggests that such gestures are inadequate and should not be employed for large interactive display environments.

The principal axis evaluation intends to evaluate the accuracy of arm orientation from XZ and YZ projection images. The result of the horizontal experiment suggests that the best performance occurs at 60 and 120 degrees. In other words, the arms are not directly pointing along the normal of the camera plane, and not too close to the user's frontal plane. The result of the vertical experiment shows that the accuracy was the worst when the user's pointing direction was closer to the camera, which was located above the display screen, 150 degrees to the participant.

6. ISO 9241-9 Standard Evaluation for Pointing

6.1 Introduction

This chapter describes the evaluation study for pointing using the ISO 9241-9 standard. Note that this evaluation is not the main objective of this research, but rather to indicate the usability of using arms for large screen interactions and most importantly, to provide quantitative results for comparisons in future research.

6.2 Evaluation for Pointing

Since the beginning of the Apple Macintosh in 1984, graphical user interfaces (GUIs) have evolved and matured. The key feature of modern GUIs is the ability for users to interact with simple point-and-select operations. The most common pointing device in desktop systems is the mouse. To select an on-screen target with a mouse, a user manipulates the mouse to manoeuvre the cursor to a target, and then selects the target by pressing and releasing a button. With an upper-limb tracker system, the user locates the target by physically pointing at it with arms; furthermore, selection would be carried out with a simple hand gesture, or if the pointing position remains stationary for a period[150].

All pointing devices are not created equal nor will they perform equally. The evaluation of a device's performance is problematic since it involves human subjects. Although there are many published evaluations of pointing devices, the methodologies are *ad hoc*. Experimental procedures are inconsistent from one study to the next, and this is an obstacle for the

ability to understand or generalize results, or to undertake between-study comparisons.

Fortunately, there is a standard from the International Standards Organization that addresses this particular problem. The full standard is ISO 9241, *Ergonomic design for office work with visual display terminals (VDTs)*. The standard is in seventeen parts. Part 9 of the standard is called *Requirements for non-keyboard input devices*. ISO 9241-9 describes a battery of tests to evaluate computer pointing devices. The procedures are well laid out and, if followed, will result in a strong and valid performance evaluation of one or more pointing devices.

Although considerable research exists in upper-limb tracking, very little has evaluated upper-limb tracking [150, 151] with the *ISO 9241 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices*. This research intends to provide quantative results for future studies by conducting an experiment based on ISO-9.

6.2.1 Throughput

ISO 9241-9 establishes uniform guidelines and testing procedures for evaluating computer pointing devices. The metric for comparison is *Throughput*, in bits per second (bits/s), which includes both the speed and accuracy of users' performances. The equation for throughput is Fitts' Index of Performance except using an *effective* index of difficulty (*IDe*). Specifically,

$$Throughput = IDe / MT \qquad (18)$$

where *MT* is the mean movement time, in seconds, for all trials within the same condition, and

$$IDe = \log_2(D/We + 1) \qquad (19)$$

IDe, in bits, is calculated from *D*, the distance to the target, and *We*, the effective width of the target. *We* is calculated as

$$We = 4.133 \times SD \qquad (20)$$

where *SD* is the standard deviation in the selection coordinates measured along the line from the centre of the home square to the centre of a target. Using effective width allows throughput to incorporate the spatial variability in human performance. It includes both speed and accuracy [152].

ISO 9241-9 was in Draft International Standard form in 1998 and became an International Standard in 2000. If one considers mouse evaluations in research not following the standard, throughput ranged from about 2.6 bits/s to 12.5 bits/s. On the contrary, studies conforming to the standard reported throughput from about 3.7 bits/s to 4.9 bits/s [153]. The data appear much more uniform and consistent. In short, ISO 9241-9 improves the quality and comparability of device evaluations.

By following the standard and comparing throughput for upper-limb tracking with a baseline technique (i.e., a mouse), it can be determined how good an upper-limb tracking system is. This is one of the few upper-limb evaluations conforming to ISO 9241-9.

An experiment was designed to implement the performance and comfort elements of ISO 9241-9. Effort was not tested since the sophisticated equipment necessary was not available for measuring biomechanical load.

Performance testing was limited to pointing and selecting using select tasks following ISO 9241-9 [154].

6.2.2 Participants

Twelve volunteer participants (eight male, four female) were recruited from the local university campus. Participants ranged from 23 to 29 years (*mean* = 27). All were daily users of computers, reporting three to twelve hours usage per day (*mean* = 7). None had prior experience with arm pointing.

6.2.3 Apparatus

The experiment environment was based on the experiment system described in 6.2.1. The bumblebee stereo camera was mounted 1.8 metres above the ground. Output was presented on a 19" monitor 0.9 metres above the ground and directly under the camera. Input was via the user's arm pointing directions.

6.2.4 Methodology

A simple point-select task was used, conforming to the multi-directional point-select test in ISO 9241-9 [154]. In this experiment each of the participants was asked to perform a "pointing and selecting" task (see the ISO standard) using their arm. A sketch of the display for the task is shown in Figure 47. The task was designed to exercise in different directions of movement and required the user to select targets arranged in a circular pattern from the centre of the screen. The targets had to be selected in sequential order as shown in figure 47. A picture of the real experimental environment is shown in figure 48.

The number of targets for the sequence in this experiment was low compared to the task proposed in the standard (20 targets). The number

was reduced to consider the greater stress and physical effort involved in performing a selection by lifting the arm, as compared to a conventional pointing device such as a mouse or a pen.

Participants were given a warm up block to familiarise themselves with arm pointing before the experiment. The experiment consisted of blocks of five sequences each with five trials. Selecting the target marked '0' started a block of trials. Each trial started after the selection of the current target, and ended at the selection of the next target. The movement time was measured on a per trial basis. Data collection began with the first selection, thus data were not collected for the target '0'. The duration of the experiment was 20 minutes.

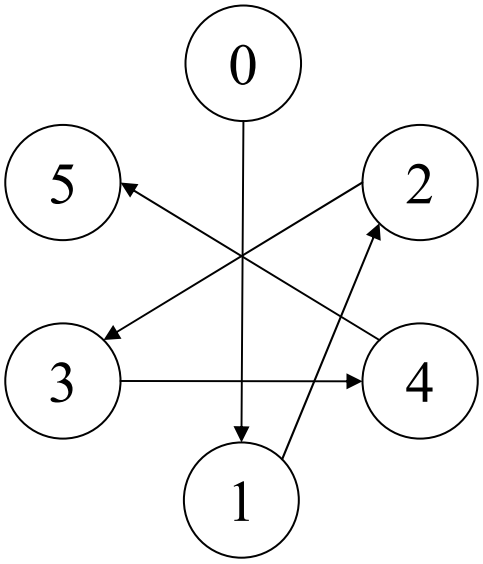


Figure 47. The target positions for the multi-directional experiment. They are arranged in a circular pattern and must be selected in sequential order 0-5.

There were six circular targets arranged in a circular layout, two metres in front of the user. The diameters of the layout circles were 200, 250, 300, 350 and 400 pixels. The target diameter was 60 pixels. A green

circle with the diameter of 10 pixels was used as the cursor for detection. The selection was based on the 'Point and Wait' suggested in [151]. The strategy consisted of pointing directly to the target on the screen, and holding the hand still for a short time period to make a selection. If the hand was detected to be motionless for 20 milliseconds, the selection was made at the centre of the cursor. To detect that the hand was not moving, position coordinates were stored and used to compare with the current position of the hand. If the variance of the position was below a certain threshold Γ, the hand was said to be still.

Figure 48. Experimental environment.

At the end of the experiment, participants were asked to respond to a written questionnaire asking them to rate their experience in using the device. The questionnaire based on [155] consisted of 13 questions covering issues of physical operation, fatigue and comfort, speed and accuracy, and overall usability. Participants were asked to respond to

each question on a 5 scale rating from low to high. The device assessment questionnaire is included in appendix B.

6.2.5 Result and Discussion

Since no other interaction methods were involved in this experiment, this research focused on the throughput of the proposed system. As mentioned in 6.2.4, the target width of the experiment was 60 pixels and the distance between targets was 200, 250, 300, 350, and 400. So according to equation (19), the indices of difficulty are 2.12, 2.37, 2.58, 2.77, and 2.94 respectively.

Based on the selection time, the throughput along different indices of difficulty is calculated using equation (18). They are: 0.0513, 0.0471, 0.0452, 0.0449, and 0.0440 bits/s. The overall average throughput is 0.0465 bits/s. Figure 49 shows the bar graph for throughput.

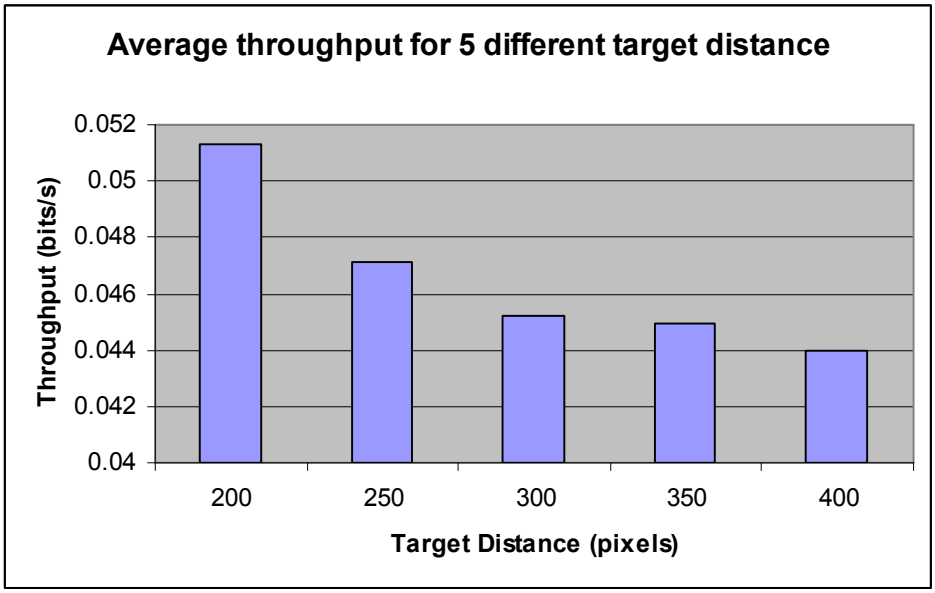


Figure 49. Result bar graph for throughput.

Comparing results to the throughput of the conventional mouse (2.6 bits/s to 12.5 bits/s) and other pointing devices: Tracker ball: 3.0 [156], Joystick: 1.8 [156], Laser pointer: 3.04 [157]. The performance of the proposed system is far from perfect. It is greatly affected by the speed of the system which runs at four frames per second. If a pointing system is slow, then the time required to point and select an object would be longer. Nevertheless, note that when passers-by are confronted with a large interactive display, they do not normally have the luxury of mouse, tracker ball, joysticks and laser pointers for interaction, but their own arms.

The device assessment questionnaire consisted of 13 questions, covering issues of physical operation, fatigue and comfort, speed and accuracy, and overall usability. Each response was rated on a 5-point scale, with the right most point the most favourable, and the left most point the least favourable response. The results are shown in figure 50.

According to the questionnaire result, participants generally felt that it was reasonably smooth during the experiment. There was little neck and shoulder fatigue, and no finger or wrist fatigue. The physical and mental effort required for the experiment was not too high. However, most of the participants felt that accurate pointing was difficult, and the operation speed was far too slow. The comment section following the questionnaire revealed that participants felt it was intuitive and interesting using arms for pointing as opposed to the conventional computer mouse. However one participant suggested that holding the arm up for more than 20 second is tiring especially having to hold the position still to allow selections.

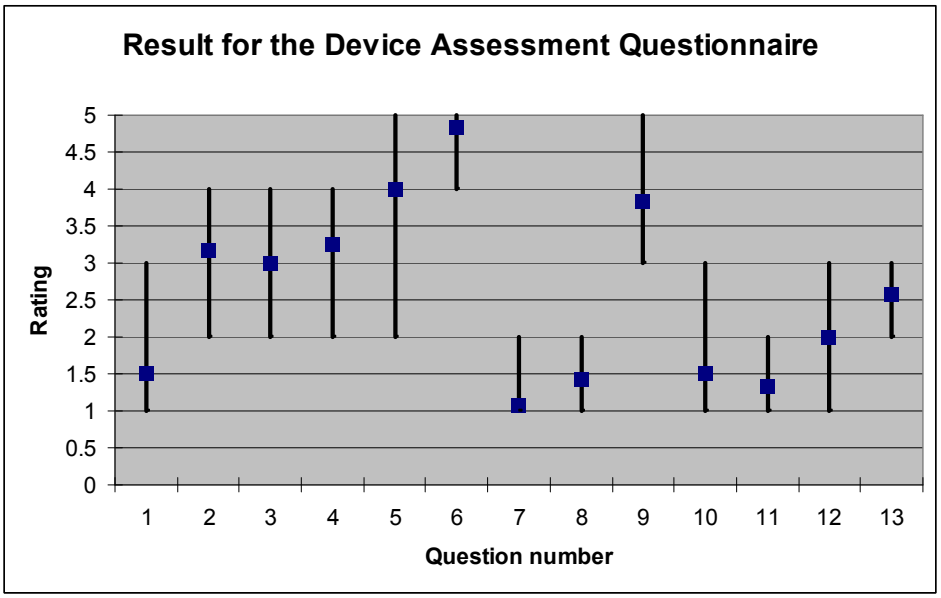


Figure 50. Result of the Arm Pointing System device assessment questionnaire. The vertical bars show the maxima and minima of user responses and the blue boxes show the average subjective ratings for each of the 13 questions.

6.3 Summary

This chapter evaluates the usability for arm pointing in large screen interactions using the ISO 9241-9 standard. In terms of multi-directional selection tasks, the throughput result shows that the proposed arm-pointing method for screen interaction is far from ideal as opposed to traditional mice. The questionnaire also reveals there were considerable amounts of mental stress and arm fatigue for screen interactions using arm motions. Despite these shortcomings, almost all participants in the survey responded that the overall experience of arm interaction was exciting and they felt that physically pointing with their arms was natural and intuitive.

7. Conclusions and Future Work

7.1 Conclusions

This research successfully evaluated a proposed algorithm for upper limb interactions with a very large interactive display. An optimal set of interactions with such a display was also proposed and shown to be most useful. Finally the usability constraints of the proposed algorithm and set of interactions were also successfully evaluated.

This thesis proposed a new algorithm which combines stereopsis, a cascade of classifiers for face recognition, anthropometric constraints, median filters, orthogonal projections, closest component matching algorithm and closed world tracking to robustly track upper limb pose and movements in unconstrained environments. The main novelty of the proposed algorithm is the new use of orthogonal projection images derived from projecting 3D point cloud onto three orthogonal planes to derive robust upper limb pose and pointing in relation to face position. As a result, three silhouette views are created and used for user pose recognition and tracking in the proposed system.

Several experimental systems were implemented using the proposed algorithm for evaluation. These included the pointing (deictic gestures) system for both usability and performance experiments; systems which track direct manipulative gestures and recognize flag semaphoric gestures; and systems which track the principal axis of upper limbs.

The performance of the pointing experiment showed that the total average error was 0.33 m and the maximum error was 0.43m. Compared

to prior research, the system proposed by this research is more robust, particularly when the target for pointing is positioned between 1.4 to 1.6m above the ground.

According to experimental results for direct manipulative gestures, grab (1 hand) and grab (2 hands) had the highest number of correct detections. They both had 94% of accuracy. Input such as drag (1 hand), drag (2 hands), pinch open, pinch close, double grab (1 hand), and double grab (2 hands) had considerably good accuracy ranging from 72% to 86%. In contrast, rotate clockwise and rotate anti-clockwise had the worst accuracy of 56% and 60% respectively. Despite the fact that there has been no prior research that emulates the complete iPod touch / iPhone direct manipulative gestures using human upper limbs, there were experiments conducted for hand grab, zoom in and zoom out. Comparing the result with prior research shows that the proposed system has at least the same or better accuracy.

A flag semaphore experiment and principal axes experiments were conducted to evaluate the accuracy for detecting arms in different orientations. The results show that the best performance occurred when the user pointed horizontally at 60 and 120 degrees directions and vertically at 60, 75 and 105 degree directions. The flag semaphore experiment result showed that the worst performance occurred when the user had to perform unnatural gestures which required the hand to be brought across the body so that both hands were on the same side.

According to the ISO 9241-9 evaluation for pointing in chapter 6, arm pointing is far less usable compared with the traditional mouse, tracker ball and joystick. However, the purpose of this research is not to replace the mouse on a traditional workstation, but to enable interaction with a

very large display. Also derived from device questionnaires, participants responded that even though arm pointing required more physical and mental effort, it was very intuitive and interesting as opposed to the conventional computer mouse.

In summary, the proposed algorithm is designed to fulfil the research motivations of:

- Obtaining arm orientation (principal axis)
- Tracking and recognizing arm and hand poses

The proposed algorithm is also robust for illumination changes, back ground and clothing. As part of the research interest, an optimal set of interactions were proposed for large interactive displays. The proposed algorithm was also used to evaluate direct manipulative gestures using arm orientations and hand gestures, revealing the opportunity of providing more complex interactions for large interactive displays.

The results suggest that the proposed algorithm and optimal set of interactions are useful for interacting with very large displays.

7.2 Future Work

Future research will pursue optimising the following aspects:

- In this research, only upper limb tracking is considered. Further research will be carried out to extend the current upper limb tracking to whole body tracking using robust multiple projection images of the whole body from multiple stereo cameras.

- The system proposed in this research tracks arm movements and recognizes hand gestures. Therefore, a natural extension of this research is using the proposed algorithmic approach for recognizing sign language.

Appendix A

Skin detection using different colour spaces

Linear and non-linear RGB spaces

In the literature [158, 159] RGB spaces are usually grouped into linear and non-linear. Linear RGB space means that it is linear to the intensity, whereas a non-linear RGB space (R'G'B') is non-linear to the intensity. Many cameras have build-in gamma correction and several image storage formats are storing gamma corrected RGBs, e.g., JPEG and GIF. Linear RGB is used, for example in computer graphics [159] and in computer vision.

All the following colour spaces are transformations from linear or non-linear RGB space respectively. Furthermore, only the intensity invariant components of the colour spaces will be shown in the following.

Normalised RGB.

A non-linear transformation of the RGB space are normalised rgb. They are obtained by normalising the colour elements (*R; G;B*) of linear RGB with their first norm:

$$r = \frac{R}{R+G+B},\ g = \frac{G}{R+G+B},\ b = \frac{B}{R+G+B}$$

Figure 51 shows the rg values of the modelled skin colours from linear (left) and non-linear (right) RGB, respectively. The *gamut* (all possible colours of the colour space) is within the triangle.

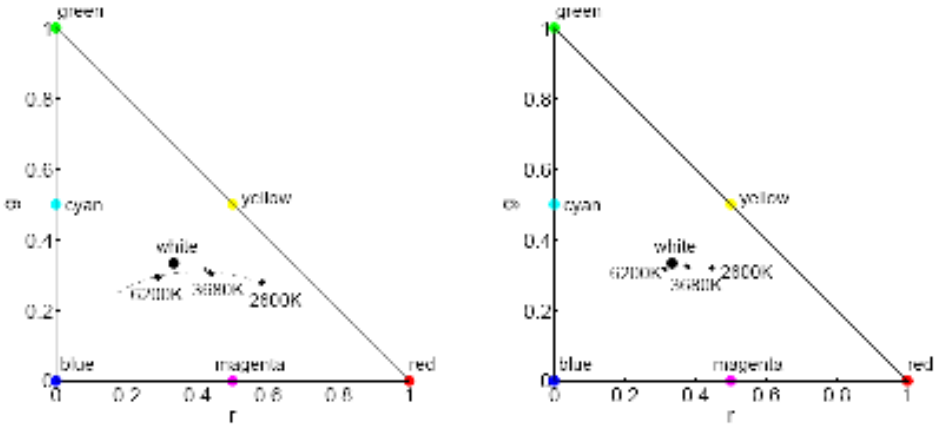


Figure 51. Rg-chromaticity plane. Left: Linear RGB. Right: Non-linear RGB.

In Figure 51 it can be seen that the skin chromaticities of the three illuminant colours are different and that the skin chromaticities of one illuminant colour under different intensities fall nearly on the same point. The dashed line shows the skin chromaticities for illuminations ranging from 2600 to 25000K.

HSI Family.

The hue-saturation-intensity family represents user-oriented colour spaces, which are primarily used in computer graphics. There are several versions: HSI, HSV, HSB, HSL. They differ mainly in the calculation of the last component, which might be the linear intensity (I) or the non-linear lightness (L). The conversion from RGB to the HSI family is non-linear. An example from RGB to HSV is given below, others may be found in [158].

$$H_1 = \arccos\left(\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right)$$

$$H = \begin{cases} H_1 & ,if \quad B \le G \\ 360° - H_1, & if \quad B > G \end{cases}$$

$$S = \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)}$$

$$V = \frac{\max(R,G,B)}{255}$$

Although the HSI spaces were originally made for user interaction they have also been used in computer vision for intensity invariant colour based segmentation by using the hue-saturation plane. Figure 52 shows the modelled skin colours in the HS-plane.

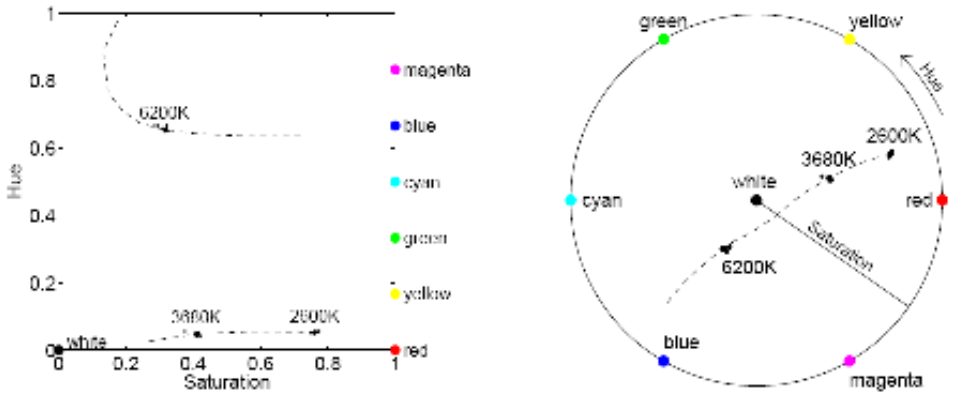


Figure 52. HS-plane of HSV space. Left: Cartesian coordinates. Right: Cartesian coordinates but using HS as polar coordinates.

The HS values of skin under one illuminant colour cluster on the same point, apart from quantisation noise, as above for the rg-chromaticities and RGB ratios. The gamut is within the square (left) or circle (right), respectively. It can be seen in the figures that when using HS as polar coordinates skin under different illuminants falls approximately along a line.

Colour difference coding spaces.

Colour difference coding spaces consist of three components: Non-linear $R'G'B'$ is transformed into luminance Y and two colour difference

components *B-Y* and *R-Y*. Examples are Y'CBCR for digital video and Y'PBPR for analogue video, also YUV, YES, and YIQ. The transformation is linear.

$$\begin{bmatrix} Y'_{601} \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.0 \\ 112.0 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

Figure 53 shows the modelled skin colour for the CBCR-plane of the Y'CBCR space (left) and the ES-plane of the YES space (right).

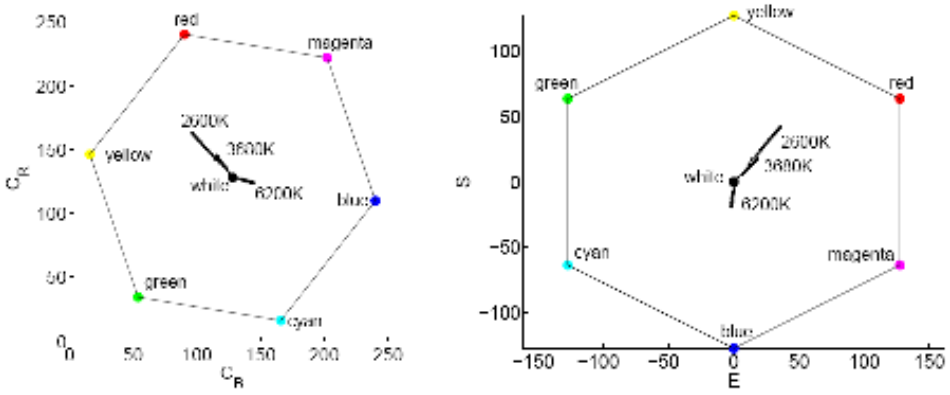


Figure 53. Left: C_BC_R-plane of Y'C_BC_R space. Right: ES-plane of YES space.

The hexagons show the gamut. It can be seen that these spaces are not as invariant to intensity changes as the above described colour spaces. Note that therefore no skin-locus (dashed line in the other figures) is shown in these figures.

LUX colour space.

Lievin and Luthon [160] recently suggested a new nonlinear colour

space for skin detection called Logarithmic hUe eXtention. This colour space derived from a colour difference coding space and is calculated as follows:

$$L = (R+1)^{0.3}(G+1)^{0.6}(B+1)^{0.1} - 1$$

$$U = \begin{cases} \frac{M}{2}(\frac{R+1}{L+1}), & if \quad R < L \\ M - \frac{M}{2}(\frac{L+1}{R+1}), & otherwise \end{cases}$$

$$X = \begin{cases} \frac{M}{2}(\frac{B+1}{L+1}), & if \quad B < L \\ M - \frac{M}{2}(\frac{L+1}{R+1}), & otherwise \end{cases}$$

where M is the dynamic range, e.g., for 8Bit data the range is [0; 255] and M = 255. UX are the chroma components, see figure 54. It can be seen that the UX values of skin under one illuminant colour cluster around the same point. In [122] it was shown that this colour space provides more contrast between skin, lips, and other materials than CBCR.

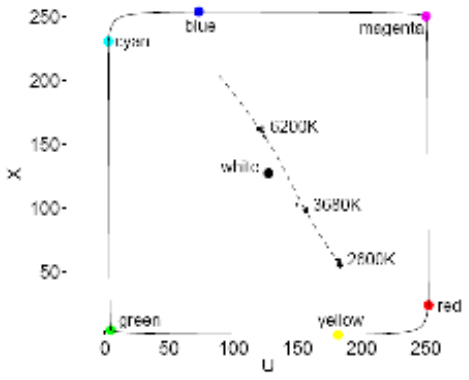


Figure 54. UX components of LUX space.

Opponent colour spaces.

The opponent colour spaces are also inspired by the human visual system that may be expressed in terms of the two opponent hues, yellow-blue

and green-red. A log-opponent colour space for skin colour detection was suggested in Fleck et al. [161]:

$$L(x) = 105\log_{10}(x+1+n)$$

$$I = L(G)$$

$$R_g = L(R) - L(G)$$

$$B_y = L(B) - \frac{L(G)+L(R)}{2}$$

where n is a random noise value generated from a distribution uniform over the range (0*;* 1) and the constant 105 is used to scale the range to the interval [-255*;* 255]. Figure 55 shows the modelled skin colours in log-opponent colour space. As with the rg-chromaticities and the HS-plane the skin colours under one illuminant colour fall on approximately the same point.

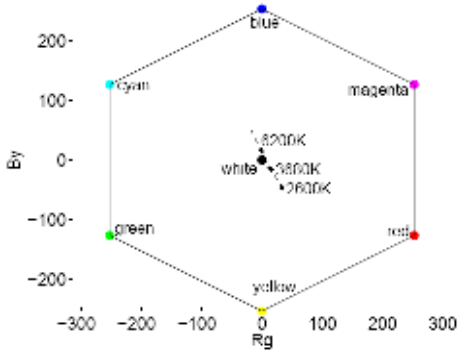


Figure 55. Log-opponent plane.

TSV and TSL spaces.

Terrillon et al. [162, 163] suggested the TSV [162] and TSL [163] space, respectively. T is the tint, S the saturation, V the value as in HSV, and L the luminance for gamma-corrected RGB values. T and S are based on normalised RGB, thus, a similar performance in intensity normalisation to rg-chromaticities is expected. The TSV and TSL colour spaces are

defined as follows, and figure 56 shows the modelled skin colours in log-opponent colour space.

$$S = \sqrt{9/5(r'^2 + g'^2)}$$

$$T = \begin{cases} \arctan(r'/g')/2\pi + 1/4, & g' > 0 \\ \arctan(r'/g')/2\pi + 3/4, & g' < 0 \\ 0, & g' = 0 \end{cases}$$

$$V = (R+G+B)/3 \quad or \quad L = 0.299R + 0.587G + 0.144B$$

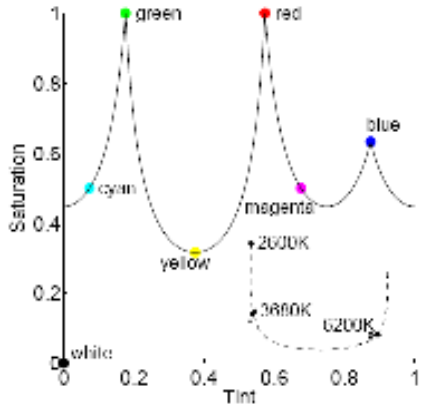


Figure 56. Tint-Saturation plane of TSV space.

CIE Colour Spaces.

The CIE (*Commission Internationale de L' Eclairage* or *International Commission on Illumination*) is the international body responsible for standards in the area of colour perception. The CIE has standardised colourimetric and perceptual uniform colour spaces, e.g., CIE XYZ, CIE L*a*b*, and CIE L*u*v*.

The CIE spaces are of minor interest in machine segmentation and recognition, particularly perceptual uniform spaces because cameras may have a higher colour resolution than humans.

Appendix B

Questionnaire used in ISO 9241-9 Evaluation.

DEVICE ASSESSMENT

Please circle the x that is most appropriate as an answer to the given comment.

1. The force required for actuation was

x	x	x	x	x
too low				too high

2. Smoothness during operation was

x	x	x	x	x
too rough				very smooth

3. The mental effort required for operation was

x	x	x	x	x
too low				too high

4. The physical effort required for operation was

x	x	x	x	x
too low				too high

5. Accurate pointing was

x	x	x	x	x
too easy				too difficult

6. Operation speed was

x	x	x	x	x
too fast				too slow

7. Finger fatigue

x	x	x	x	x
none				very high

8. Wrist fatigue

x	x	x	x	x
none				very high

9. Arm fatigue

x	x	x	x	x
none				very high

10. Shoulder fatigue

x	x	x	x	x
none				very high

11. Neck fatigue

x	x	x	x	x
none				very high

12. General comfort

x	x	x	x	x
very uncomfortable				very comfortable

13. Overall, the input device was

x	x	x	x	x
very difficult to use				very easy to use

Comments

8. References

1. H. Yoshimoto, N.D.a.S.Y. *Vision-based real-time Motion Capture system using Multiple Cameras*. in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. 2003.
2. Velastin, L.M.F.a.S.A. *People tracking in surveillance applications*. in *2nd IEEE International Workshop on Performance Evaluation on Tracking and Surveillance*. 2001. Kauai (Hawaii-USA).
3. Leubner, C.B., C. Muller, H. . *Computer-Vision-Based Human-Computer Interaction with a Back Projection Wall Using Arm Gestures*. in *Euromicro Conference 27th*. 2001.
4. Antonio Camurri, G.V., *Gesture-Based Communication in Human-Computer Interaction: 5th* 2004: springer.
5. *Gestures for Direct Manipulation*. [cited; Available from: http://www.interactivegestures.com/index.php?title=Main_Page.
6. *Hand and Arm Signals*. [cited; Available from: http://www.airsoftgent.be/dbase/hands.htm.
7. *Flag semaphore*. [cited; Available from: http://en.wikipedia.org/wiki/Semaphore_flag_signalling.
8. Croft, J. *Semaphore Flag Signalling System*. [cited; Available from: http://www.anbg.gov.au/flags/semaphore.html.
9. *British Sighn Language*. [cited; Available from: http://www.britishsignlanguage.com/.
10. *New Zealand Finger Spelling*. [cited; Available from: http://www.deaf.co.nz/index.php?option=com_content&task=view&id=52&Itemid=1.
11. *Australian Sign Language, Alphabet Key*. [cited; Available from: http://www.auslan.org.au/index.cfm?skinname=content&page=1532&contentpage=asbFingerSpell&asbFSA=1&CFID=1380421&CFTOKEN=48553424.
12. M. Harville, D.L. *Fast, integrated person tracking and activity recognition with plan-view templates from a single stereo camera*. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Washington DC.
13. M. Harville, D.L. *Stereo Person Tracking with Short and Long Term Plan-View Appearance Models of Shape and Color*. in *Proceedings of the IEEE International Conference on Advanced Video and Signal based Surveillance*. 2005. Como, Italy.
14. Vassilis Athitsos, S.S. *Estimating 3D Hand Pose from a Cluttered Image*. in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2003.

15. Cohen, I.L., H. *Inference of human postures by classification of 3D human body shape*. in *IEEE. Analysis and Modeling of Faces and Gestures. AMFG*. 2003.
16. Rosales, R.S., M. Alon, J. Sclaroff, S. *Estimating 3D body pose using uncalibrated cameras*. in *Computer Vision and Pattern Recognition. CVPR*. 2001.
17. N.Jojic, M.T.a.T.H. *Tracking Self-Occluding Articulated Objects in Dense Disparity Maps*. in *Conference on Computer Vision*. 1999. Corfu, Greece.
18. Jojic, N.T., M. Huang, T.S. *Fast Pose Estimation with Parameter-Sensitive Hashing*. in *ICCV, the Ninth IEEE International Conference on Computer Vision*. 1999. Kerkyra, Greece.
19. Matheen Siddiqui, G.M. *Robust real-time upper body limb detection and tracking*. in *International Multimedia Conference archive, the 4th ACM international workshop on Video surveillance and sensor networks*. 2006. Santa Barbara, California, USA.
20. IdeaBoard, M. [cited; Available from: http://www.3m.com/.
21. Pinhanez., C. *The everywhere displays projector: A device to create ubiquitous graphical interfaces*. in *Proceedings of Ubiquitous Computing (Ubicomp)*. 2001.
22. Ringel, M.a.H.B., Y. J., Terry Winograd. *Barehands: Implement-Free Interaction with a Wall-Mounted Display*. in *CHI '01 extended abstracts on Human factors in computing systems*. 2001.
23. *NaturalPoint SmartNav*. [cited; Available from: http://www.naturalpoint.com/smartnav/.
24. T.J. Cham, J.R., G. Sukthankar, R. Sukthankar. *Shadow elimination and occluder light suppression for multi-projector displays*. in *CVPR*. 2001.
25. C. Sminchisescu, B.T. *Covariance scaled sampling for monocular 3D body tracking*. in *CVPR*. 2001.
26. H. Sidenbladh, M.J.B., D.J. Fleet. *Stochastic tracking of 3D human figures using 2D image motion*. in *ECCV*. 2000.
27. C. Bregler, J.M. *Tracking people with twists and exponential maps*. in *CVPR*. 1998.
28. S. Wachter, H.H.N. *Tracking persons in monocular image sequences*. in *Comput. Vis. Image Understand*. 1999.
29. D.M. Gavrila, L.S.D. *3D model-based tracking of humans in action: a multi-view approach*. in *CVPR*. 1996.
30. I.A. Kakadiaris, D.M. *3D human body model acquisition from multiple views*. in *ICCV*. 1995.
31. I.A. Kakadiaris, D.M. *Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection*. in *CVPR*. 1996.

32. J. Deutscher, A.B., I. Reid. *Articulated body motion capture by annealed particle filtering*. in *CVPR*. 2000.
33. J. Deutscher, A.D., I. Reid. *Automatic partitioning of high dimensional search spaces associated with articulated body motion capture*. in *CVPR*. 2001.
34. J. Carranza, C.T., M. Magnor, H.P. Seidel. *Free-viewpoint video of human actors*. in *ACM SIGGRAPH*. 2003.
35. C. Theobalt, J.C., M. Magnor, H.P. Seidel. *Enhancing silhouette-based human motion capture with 3D motion fields*. in *Pacific Graphics*. 2003.
36. K.M. Cheung, T.K., J.-Y. Bouguet, M. Holler. *A real time system for robust 3D voxel reconstruction of human motions*. in *CVPR*. 2000.
37. K.M. Cheung, S.B., T. Kanade. *Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture*. in *CVPR*. 2003.
38. I. Mikic, M.T., E. Hunter, P. Cosman. *Articulated body posture estimation from multi-camera voxel data*. in *CVPR*. 2001.
39. I. Mikic, M.T., E. Hunter, P. Cosman. *Human body model acquisition and tracking using voxel data*. in *IJCV*. 2003.
40. J.R. Mitchelson, A.H. *Simultaneous pose estimation of multiple people using multiple-view cues with hierarchical sampling*. in *BMVC*. 2003.
41. R. Plaenkers, P.F. *Model-based silhouette extraction for accurate people tracking*. in *ECCV*. 2002.
42. Hjelmas, E.a.L., B.K. *Face detection: A survey*. in *Computer Vision and Image Understanding*. 2001.
43. M.-H. Yang, D.J.K., and N. Ahuja, *Detecting faces in images: A survey.* Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2002. **vol. 24**(no. 1): p. pp. 34-58.
44. Foley JD, v.D.A., Feiner SK & Hughes JF, *Computer Graphichs Principles and Practice: Second Edition in C*. 1996, New York.
45. M, B., *Face colour under varying illumination – analysis and applications*, in *Department of Electrical and Information Engineering, University of Oulu Infotech Oulu, University of Oulu*. 2002, University of Oulu.
46. Pentland, A.A.a.A. *Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features*. in *Proceedings of 13th ICPR*. 1996.
47. T.Darrell, G.G., M. Harville and J. Woodfill. *Integrated Person Tracking Using Stereo, Colour, and pattern detection*. in *Proceedings of the Conference on Computer vision and pattern recognition*. 1998.
48. Colombo, C., Bimbo, A. D. and Valli, A., IEEE Transactions on systems, man, and cybernetics, 2003. **33**: p. 677-686.

49. Nickel, K.a.S., R. *Pointing Gesture Recognition based on 3D-Tracking of Face, Hands and Head-Orientation*. in *Proceedings of ICMI '03 International Conference on Multimodal Interfaces*. 2003.
50. Javier Varona, José M. Buades and Francisco J. Perales. *Hands and face tracking for VR applications*. in *Computers & Graphics*. 2005.
51. Kim., S.J.J.P.C.P.I.-K.J.Y.-O.K.B. *Morphological approach of stereo camera based human motion capture system*. in *ICCAS international conference on control, automation and systems*. 2007.
52. Rafael Muñoz-Salinas, M.G.-S., Rafael Medina Carnicer, *Adaptive multi-modal stereo people tracking without background modelling.* Journal of Visual Communication and Image Representation, 2008. **19**(2): p. 75-91.
53. Gaile Gordon, X.C., Ron Buck. *Person and Gesture Tracking with Smart Stereo Cameras*. in *SPIE: Three-Dimensional Image Capture and Applications*. 2008.
54. Urmson, C., *A Comparison of Pt. Grey Research's Digiclops and Videre Design's Small Vision System for Sensing on the Hyperion Robot*. 2000.
55. Y. Guo, G.X.a.S.T. *Understanding Human Motion Patterns*. in *ICPR*. 1994.
56. M.K. Leung and Y.H. Yang. *First Sight: A Human Body Outline Labeling System*. in *IEEE Trans. on PAMI*. 1995.
57. Nelson, R.P.a.R. *Low Level Recognition of Human Motion*. in *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*. 1994.
58. GRogez, G., Rius, I., Martinez-del-Rincon, J., *Exploiting Spatio-temporal Constraints for Robust 2D Pose Tracking*, in *Second Workshop on Human Motion*. 2007, Springer.
59. Karaulova, J., Hall, P.M., Marshall, A.D. *hierarchical model for tracking people with a single video camera*. in *British Machine Vision Conference (BMVC'2000)*. 2000.
60. C. R. Wren, B.P.C., and A. P. Pentland. *Understanding purposeful human motion*. in *The Fourth International Conference on Automatic Face and Gesture Recognition*. 2000. Grenoble, France.
61. Y. Iwai, K.O., and M. Yachida. *Posture estimation using structure and motion models*. in *Int. Conf. Comput. Vision*. 1999. Corfu, Greece.
62. C. Yaniz, J.R., and F. Perales. *3D region graph for reconstruction of human motion*. in *Workshop on Perception of Human Motion at ECCV*. 1998.

63. Schrotter, G. *Realistic Body Modeling out of Video Sequences: First Application to Body Parts*. in *SPIE-IS&T Electronic Imaging*. 2005. San Jose, USA.
64. Remondino, F.R., A. *Human figure reconstruction and modeling from single image or monocular video sequence*. 2003.
65. Christian Theobalt , M.M., Pascal Schüler , Hans-Peter Seidel. *Combining 2D Feature Tracking and Volume Reconstruction for Online Video-Based Human Motion Capture*. in *10th Pacific Conference on Computer Graphics and Applications*. 2002.
66. Francesc Moreno-Noguer, A.S., Dimitris Samaras. *Integration of deformable contours and a multiple hypotheses Fisher color model for robust tracking in varying illuminant environments*. in *Image and Vision Computing*. 2007.
67. Panin, G.L., A.; Knoll, A. *An Efficient and Robust Real-Time Contour Tracking System*. in *IEEE International Conference on Computer Vision Systems*. 2006.
68. Ju, S., Black, M, Yaccob, Y. *Cardboard people: a parameterized model of articulated image motion*. in *IEEE Int. Conf. on Automatic Face and Gesture Recognition*. 1996.
69. Y. Kameda, M.M., and K. Ikeda. *Three dimensional motion estimation of a human body using a difference image sequence*. in *Asian Conference on Computer Vision*. 1995.
70. C. Hu, Q.T., Y. Li, and S. Ma. *Extraction of parametric human model for posture recognition using genetic algorithm*. in *The fourth International Conference on Automatic Face and Gesture Recognition*. 2000. Grenoble, France.
71. Medioni, C.Y.a.G. *Inferring 3D Volumetric Shape of Both Moving Objects and Static Background Observed by a Moving Camera*. in *IEEE Conference on Computer Vision and Pattern Recognition*. 2007.
72. Urtasun, R., Fleet, D., Fua, Pascal. *3D People Tracking with Gaussian Process Dynamical Models*. in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2006.
73. Kehl, R.B., M.; Van Gool, L. *Full Body Tracking from Multiple Views Using Stochastic Sampling*. in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2005.
74. Q. Delamarre, O.D.F. *3D articulated models and multi-view tracking with silhouettes*. in *ICCV*. 1999.
75. Urtasun, R.F., P. *3D tracking for gait characterization and recognition*. in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004*. 2004.
76. Luck, J., Small, D., Little, C., *Real-time tracking of articulated human models using a 3d shape-from-silhouette method*. Lecture notes in computer science. 2001.

77. Er Ladikos, S.B., Nassir Navab. *A REAL-TIME TRACKING SYSTEM COMBINING TEMPLATE-BASED AND FEATURE-BASED APPROACHES*. in *VISAPP*. 2007.
78. D. Jang, H.C. *Active models for tracking moving objects*. in *Pattern Recogn. 33*. 2000.
79. Sclaroff, R.R.a.S. *Learning and synthesizing human body motion and posture*. in *Fourth International Conference on Automatic Face and Gesture Recognition*. 2000. Grenoble, France.
80. G. Stamou, M.K., E. Loutas, N. Nikolaidis, and I. Pitas. *2D and 3D motion tracking in digital video*. in *Handbook of Image and Video Processing*. 2005.
81. Hieu T. Nguyen, M.W., Rein van den Boomgaard. *Occlusion Robust Adaptive Template Tracking*. in *IEEE Conf. on Computer Vision, ICCV'2001*. 2001.
82. Bhatia S., L.S., M. Isard and M. Black. *3D Human Limb Detection using Space Carving and Multi-view Eigen Models*. in *Second IEEE International Conference on Computer Vision Systems*. 2004.
83. Rehg, J.a.T.K. *Model-based tracking of self-occluding articulated objects*. in *Fifth International Conference on Computer Vision*. 1995.
84. Pentland, A.a.B.H., *Recovery of nonrigid motion and structure.* IEEE Transactions on PAMI, 1991(13): p. p 730-742.
85. Wren, C., A. Azarbayejani, T. Darrell and A. Pentland, *Pfinder: Real-time tracking of the human body.* IEEE Transactions on PAMI, 1997. **19(7)**: p. p 780-785.
86. Karam, M.a.s., m. c, *A Taxonomy of Gestures in Human Computer Interactions*, in *Technical Report ECSTR-IAM05-009*, E.a.C. Science, Editor. 2005, University of Southampton.
87. Ward, D.J., Blackwell, A. F., and MacKay, D. J. C. *Dasher - a data entry interface using continuous gestures and language models*. in *Proceedings of the ACM Workshop on Perceptive User Interfaces.* 2000: ACM Press.
88. Ou, J., Fussell, S. R., Chen, X., Setlock, L. D., and Yang, J. *Gestural communication over video stream: supporting multimodal interaction for remote collaborative physical tasks*. in *Proceedings of the 5th international conference on Multimodal interfaces.* 2003: ACM Press.
89. Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X.-f., Kirbas, C., McCullough, K. E., and Ansari, R. *Multimodal human discourse: gesture and speech*. in *ACM Trans. Human-Computer Interaction.* 2002.
90. Rekimoto, J. *Pick-and-drop: a direct manipulation technique for multiple computer environments*. in *Proceedings of the 10th*

annual ACM symposium on User interface software and technology. 1997: ACM Press.
91. Wu, M.a.B., R. *Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays*. in *Proceedings of the 16th annual ACM symposium on User interface software and technology*. 2003: ACM Press.
92. Rekimoto., J. *SmartSkin: an infrastructure for freehand manipulation on interactive surfaces*. in *CHI*. 2002.
93. Wexelblat, A. *Research challenges in gesture: Open issues and unsolved problems*. in *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*. 1998: Springer-Verlag.
94. Alpern, M.a.M., K. *Developing a car gesture interface for use as a secondary task*. in *CHI '03 extended abstracts on Human factors in computing systems*. 2003: ACM Press.
95. Baudel, T.a.B.-L., M. *Charade: remote control of objects using free-hand gestures*. in *Commun*. 1993: ACM.
96. Grossman, T., Wigdor, D., and Balakrishnan, R. *Multi-finger gestural interaction with 3d volumetric displays*. in *Proceedings of the 17th annual ACM symposium on User interface software and technology*. 2004: ACM Press.
97. Rekimoto, J., Ishizawa, T., Schwesig, C., and Oba, H. *Presense: interaction techniques for finger sensing input devices*. in *Proceedings of the 16th annual ACM symposium on User interface software and technology*. 2003: ACM Press.
98. Bolt, R.A. *Put-that-there: Voice and gesture at the graphics interface*. in *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. 1980: ACM Press.
99. Schmandt, C., Kim, J., Lee, K., Vallejo, G., and Ackerman, M. *Mediated voice communication via mobile ip*. in *Proceedings of the 15th annual ACM symposium on User interface software and technology*. 2002: ACM Press.
100. Davis, J.W.a.V., S. *A perceptual user interface for recognizing head gesture acknowledgements*. in *Proceedings of the 2001 workshop on Percetive user interfaces*. 2001: ACM Press.
101. Paradiso, J.A. *Tracking contact and free gesture across large interactive surfaces*. in *Commun*. 2003: ACM.
102. Wilson, A.a.S., S. *Xwand: Ui for intelligent spaces*. in *Proceedings of the conference on Human factors in computing systems*. 2003: ACM Press.
103. Moyle, M.a.C., A. *The design and evaluation of a flick gesture for 'back' and 'forward' in web browsers*. in *Proceedings of the Fourth Australian user interface conference on User interfaces*. 2003: Australian Computer Society.

104. *How Digital Cameras Work*. [cited; Available from: http://electronics.howstuffworks.com/digitalcamera.htm.
105. R. Jain, R.K.a.B.S., *Machine Vision*. 1995. p. 112-138.
106. Ponce, D.F.a.J., *Computer Vision, A Modern Approach*. 2003.
107. Intelli., A.F.B.a.S.S. *Large Occlusion Stereo*. in *International Journal of Computer Vision (IJCV)*. 1999.
108. Veksler., O. *Stereo Matching by Compact Windows via Minimum Ratio Cycle*. in *ICCV*. 2001.
109. Belhumeur., P.N., *A Bayesian-approach to Binocular Stereopsis.* International Journal of Computer Vision (IJCV), 1996. **19**: p. 237-260.
110. D. Geiger, B.L.a.A.Y., *Occlusions and Binocular Stereo.* International Journal of Computer Vision (IJCV), 1995. **14**(3): p. 211-226.
111. Geiger, H.I.a.D. *Occlusions, Discontinuities and Epipolar Lines in Stereo*. in *ECCV*. 1998.
112. Viola, P.a.J., M. *Rapid object detection using boosted cascade of simple features*. in *IEEE Conference on Computer Vision and Pattern Recognition*. 2001.
113. Freund, Y., and Schapire, R. E. *A decision-theoretic generalization of on-line learning and an application to boosting*. in *the second European Conference on Computational Learning Theory*. 1995. Springer-Verlag.
114. Rowley, H.A., Baluja, S., and Kanade, T. *Neural network-based face detection*. in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1998.
115. Feraud, R., Bernier, O., Viallet, J.-E., and Collobert, M. *A fast and accurate face detector based on neural networks*. in *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*. 20001.
116. Rein-Lien Hsu; Abdel-Mottaleb, M.J., A.K. *Face detection in color images*. in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2002.
117. Fleuret, F., and Geman, D., *Course-to-fine face detection.* International Journal of Computer Vision, 2001. **41**: p. 85-107.
118. Yow, K., and Cipolla, R., *Enhancing human face detection using motion and active contours* 1997, Springer Belin / Heidelberg.
119. Pitas, S.T.a.I. *Facial feature extraction in frontal views using biometric analogies*. in *European signal processing conference*. 1998.
120. Lienhart, R.a.M., J. *An extended set of Haar-like features for rapid object detection*. in *IEEE ICIP*. 2002.
121. Menezes, P., Barreto, J.C. and Dias, J. *Face tracking based on Haar-like features and eigenfaces*. in *5th IFAC Symposium on Intelligent Autonomous Vehicles*. 2004. Lisbon, Portugal.

122. Pheasant, S., *Bodyspace. Anthropometry, Ergonomics and the Design of Work*. 1996.
123. Gao, J.a.J.S., *Inferring Human Upper Body Motion Using Belief Propagation*. 2003, Robotics Institute, Carnegie Mellon University.
124. S.Carbini, J.E.V., O. Bernier, and B. Bascle, *Tracking Body Parts of Multiple People for Multi-Person Multimodal Interface*. 2005.
125. Kelvin C, M.T. *Real-time Monocular Tracking of View Frustum for Large Screen Human-Computer Interaction*. in *Twenty-Eighth Australasian Computer Science Conference*. 2005.
126. Baxes., G.A., *Digital Image Processing. Principles and Applications*. 1994.
127. Shapiro, R.M.H.a.L.G., *Computer and Robot Vision*. 1992: Addison-Wesley Longman Publishing Co., Inc.
128. Lin Yin Ruikang Yang Gabbouj, M.N., Y. *Weighted Median Filters: A Tutorial*. in *IEEE Trans. on Circuits and Systems*. 1996.
129. Cowart, W.S.a.A. *An iterative approach to region growing*. in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1983.
130. Bobick, S.S.I.J.W.D.A.F. *Real-time closed-world tracking*. in *CVPR Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. 1997.
131. Fischler, T.M.S.a.M.A. *Context-based vision: recognizing objects using information from both 2D and 3D imagery*. in *IEEE Trans. Patt. Analy. and Mach. Intell*. 1991.
132. Andrew D. Wilson. *TouchLight: An Imaging Touch Screen*. in *ICMI*. 2004.
133. Wilson., A.D. *PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System*. in *UIST*. 2005.
134. Han., J.Y. *Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection*. in *UIST*. 2005.
135. Izadi, S., Agarwal, A., Criminisi, A., Winn, J., Blake, A., Fitzgibbon. A. *C-Slate: A Multi-Touch and Object Recognition System for Remote Collaboration using Horizontal Surfaces*. in *IEEE Conference on Horizontal Interactive Human-Computer Systems*. 2007.
136. Buxton, W. *Multi-Touch Systems that I Have Known and Loved*. [cited; Available from: http://www.billbuxton.com/multitouch.
137. Dietz, P.H.L., D.L. *DiamondTouch: a multi-user touch technology*. in *ACM Symposium on User Interface Software and Technology (UIST)*. 2001.
138. *JazzMutant Lemur*. [cited; Available from: http://www.jazzmutant.com/lemur_overview.php.

139. Westerman, W., *Hand Tracking, Finger Identification and Chordic Manipulation on a Multi-Touch Surface*. 1999, University of Delaware,.
140. Inc., T.C. *Array Sensors*. [cited; Available from: http://www.tactex.com/products_array.php.
141. Wu, M.e.a. *Multi-finger and whole hand gestural techniques for tabletop displays.* in *UIST*. 2003.
142. al., P.H.D.e. *DT Controls: Adding Identity to Physical Interfaces*. in *UIST*. 2005.
143. Thomas B. Moeslund , M.S.a.E.G., *A Natural Interface to a Virtual Environment through Computer Vision-Estimated Pointing Gestures*. Lecture Notes in Computer Science. 2002: Springer Berlin / Heidelberg.
144. Martin Zobl, R.N., Michael Geiger, Manfred Lang and Gerhard Rigoll *Gesture Components for Natural Interaction with In-Car Devices*. Lecture Notes in Computer Science. 2004: Springer Berlin / Heidelberg.
145. Utsumi, A.O., J. *Multiple-hand-gesture tracking using multiple cameras*. in *Computer Vision and Pattern Recognition*. 1999.
146. Xiying W, X.Z., GuoZhong D. *Tracking of deformable human hand in real time as continuous input for gesture-based interaction*. in *12th international conference on Intelligent user interfaces*. 2007.
147. Juang W, H.S., Yael E, Michael G, Craig F, Mark S, Jon H., *Real-Time Hand Gesture Interface for Browsing Medical Images.* Journal of the American Medical Informatics Association, 2007. **15**(3).
148. Ahmad Al-Mazeed, M.N.a.S.G., *Classifiers Combination for Improved Motion Segmentation*. Lecture Notes in Computer Science. 2004: Springer Berlin / Heidelberg.
149. Serafinavičius P., D.G. *Detection of Hand Position using 3-D Computer Vision*. in *Electronics and Electrical Engineering. – Kaunas: Technologija*. 2006.
150. Fang, C., Eric, C., Julien, E., Serge, L., Natalie, R., Yu, S., Ronnie, T., Mike, Wu. *A study of manual gesture-based selection for the PEMMI multimodal transport management interface*. in *Proceedings of the 7th international conference on Multimodal interfaces*. 2005: ACM.
151. Emilio S, R.S. *Experimental evaluation of vision and speech based multimodal interfaces*. in *workshop on Perceptive user interfaces*. 2001. Orlando, Florida.
152. MacKenzie, I.S. *Fitts' law as a research and design tool in human-computer interaction*. in *Human-Computer Interaction*. 1992.

153. Soukoreff, R.W., MacKenzie, I.S., *Towards a standard for pointing device evaluation: Perspectives on 27 years of Fitts' law research in HCI.* International Journal of Human-Computer Studies, 2004. **61**: p. 751–789.
154. ISO, *ISO/DIS 9241-9 Ergonomic Requirements for Office work Work with Visual Display Terminals, Non-keyboard Input Device Requirements, Draft. International Standard, International Organization for Standardization.* 1998.
155. Douglas, S.A., Kirkpatrick, A.E., MacKenzie, I.S. *Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard*. in *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99*. 1999. New York.
156. MacKenzie, I.S., Kauppinen, T. and Silfverberg, M. *Accuracy Measures for Evaluating Computer Pointing Devices*. in *Proc CHI'01*. 2001.
157. Oh, J.-Y.a.S., W. *Laser Pointers as Collaborative Pointing Devices*. in *Proc Graphics Interface*. 2002.
158. Plataniotis, K.N.a.V., A.N. *Color image processing and applications*. in *Springer*. 2000. Berlin, Germany.
159. Poynton, C.A., *A Technical Introduction to Digital Video*. John Wiley & Sons. 1996.
160. Lievin, M.a.L., F, *Nonlinear color space and spatiotemporal mrf for hierarchical segmentation of face features in video.* IEEE Trans. on Image Processing, 2004. **13**(1): p. 63-71.
161. Fleck, M.M., Forsyth, D.A., and Bregler, C. *Finding naked people*. in *4th European Conf. on Computer Vision*. 1996. Cambridge, UK.
162. Terrillon, J.C., David, M., and Akamatsu, S. *Detection of human faces in complex scene images by use of a skin color model and of invariant fourier-mellin moments*. in *Int. Conf. on Pattern Recognition*. 1998. Brisbane, Australia.
163. Terrillon, J.C., Shirazi, M.N., Fukamachi, H., and Akamatsu, S. *Comarative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images*. in *4th IEEE Int. Conf. on Automatic Face- and Gesture-Recognition*. 2000. Grenoble, France.